

**UNIVERSIDAD DE TARAPACÁ**



**ESCUELA UNIVERSITARIA DE INGENIERÍA INDUSTRIAL,  
INFORMÁTICA Y SISTEMAS**

Área de Ingeniería en Computación e Informática



## **Informe Final - iCodeClass**

**Autores: Luis Lecaros  
Daniela Oñate  
Charlotte Soto  
Manuel Tapia  
Fabián Urrutia  
Diego Valencia**

**Asignatura: Proyecto III**

**Profesor: Ricardo Valdivia**

ARICA, 11 de Diciembre 2018

## Contenidos

SECCIÓN	PÁGINA
<b>I. INTRODUCCIÓN</b>	<b>3</b>
<b>II. DESARROLLO</b>	<b>4</b>
2.1 Especificación del problema	4
2.2 Metodología de desarrollo	5
2.2.1 Roles	5
2.2.2 Product Backlog	5
2.3 Descripción de objetivos	6
2.3.1 Objetivo General	6
2.3.2 Objetivo Específico	6
2.4 Herramientas Utilizadas	7
2.5 Especificación de Requisitos (Product Backlog)	8
2.6 Estructura organizacional	10
2.6.1 Roles	10
2.6.2 Responsabilidades Iniciales	10
2.7 Gestión de riesgos	11
2.7.1 Tabla de riesgo del proyecto	11
2.8 Costeo	12
2.9 Avance por Sprint	13
2.9.1 Sprint 1	13
2.9.1.1 Sprint Backlog	13
2.9.1.2 Diseño	14
2.9.1.3 Implementación (Vistas)	18
2.9.1.4 Aspectos Generales	19
2.9.2 Sprint 2	22
2.9.2.1 Sprint Backlog	22
2.9.2.2 Diseño	23
2.9.2.3 Implementación	34
2.9.2.4 Aspectos Generales	35
2.9.3 Sprint 3	38
2.9.3.1 Sprint Backlog	38
2.9.3.2 Diseño	39
2.9.3.3 Implementación	55
2.9.3.4 Aspectos Generales	57
<b>III. CONCLUSIONES</b>	<b>61</b>
<b>IV. REFERENCIAS BIBLIOGRÁFICAS</b>	<b>62</b>
<b>V. ANEXOS</b>	<b>62</b>

## I. INTRODUCCIÓN

Con motivo de mejorar los métodos actuales de realización de actividades académicas, principalmente en el área de computación e informática de la “Universidad de Tarapacá”, se han analizado y descubierto desventajas del método de enseñanza de la institución, en específico del área de “Computación e Informática” que pueden ser aprovechadas mediante el uso de tecnologías web.

Dicho problema se encuentra en la enseñanza de programación y la creación de código en general. El cual cuenta en la actualidad con métodos simples o primitivos que pueden ser mejorados. Es por esto que en el presente curso de “Proyecto 3” se ha establecido el desafío de implementar un sistema informático que mejore dichos métodos.

Para la implementación del sistema informático se utilizarán diversas tecnologías actuales y actualizadas. Su desarrollo será basado en un sistema ya existente y similar al que se desea implementar, cuya utilidad y recursos que provee servirán de ayuda para el desarrollo del proyecto.

El presente documento de “Informe Final - iCodeClass” presenta todas las aristas, en cuanto la organización del equipo de trabajo, objetivos del proyecto, metodologías de desarrollo, planificación del trabajo, herramientas a utilizar, riesgos y costos del proyecto. Dichos puntos fueron mejorados y actualizados luego de la presentación del Segundo Sprint del proyecto.

Para este informe final se consideran aspectos como la descripción de los requerimientos del proyecto y todo el progreso de estos hasta la fecha de finalización del proyecto en cuanto a los objetivos de cada sprint, el diseño de acuerdo a los objetivos, su implementación y aspectos generales en cuanto a los responsables de su desarrollo. Además de presentar gráficamente el avance de la planificación mediante la Carta Gantt.

## II. DESARROLLO

### 2.1 Especificación del problema

En la actualidad la enseñanza en los cursos de programación en la Universidad de Tarapacá (Sede Arica) se realiza con la siguiente metodología: El profesor presenta las cátedras sobre el contenido a enseñar y luego proporciona una serie de ejercicios a sus alumnos con motivo de que estos practiquen lo aprendido en el lenguaje de programación correspondiente. Esto último se realiza de manera simple, individual y sin ningún tipo de colaboración directa u online con el profesor o compañero, sin la utilización de herramientas que puedan proporcionar un entorno de trabajo con mayor dinamismo. Lo anterior, puede generar poco interés por parte de los alumnos en la participación de las actividades que se presentan en clases debido a que no son colaborativas.

## 2.2 Metodología de desarrollo

Se utilizará Scrum como metodología de desarrollo para el proyecto “i-Code Class”, en la siguiente ilustración (ilus. 1) se muestra un ejemplo de esta metodología.

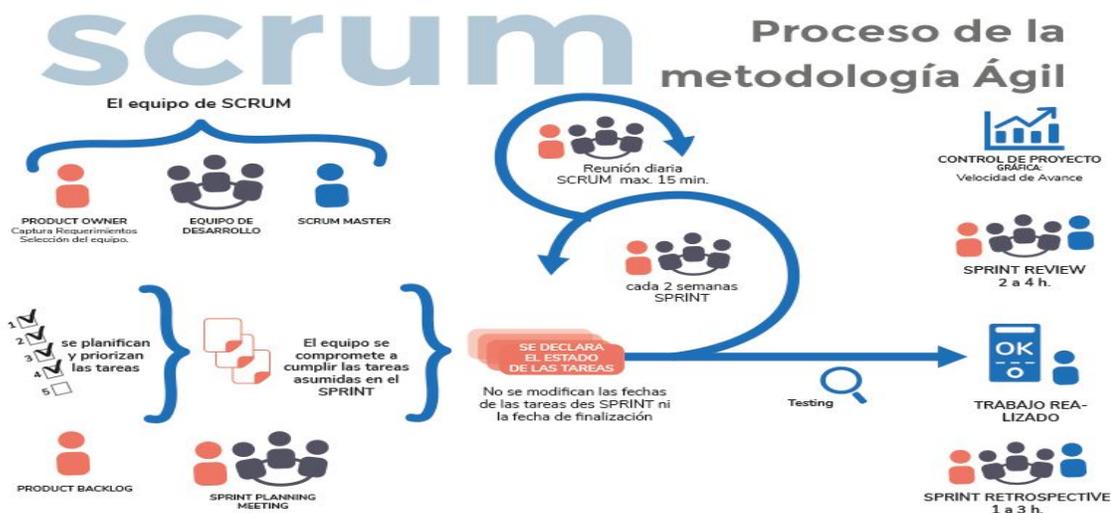


Ilustración 1. Metodología SCRUM

### 2.2.1 Roles

Los roles que tendrá el proyecto serán los siguientes:

- Product owner: Dueño del producto, el cual deberá estar estrechamente relacionado con el proyecto debido a que Scrum define una serie de reuniones para definir cuáles serán los pasos a seguir en el próximo Sprint.
- Scrum master: Quien liderará el equipo de desarrollo del proyecto, facilitando todas las herramientas, procesos y plazos de entregas para que el equipo pueda alcanzar los objetivos definidos previamente y llegar a la fase de “Sprint Final” sin mayor dificultad.
- Equipo de desarrollo: Como su nombre lo indica, serán los encargados de llevar a cabo el desarrollo del proyecto, codificando o bien documentando los avances realizados para cumplir con el Sprint Backlog.

### 2.2.2 Product Backlog

El Product Backlog es en palabras simples una lista de todas las cosas que necesitan ser realizadas para completar el proyecto, reemplazará a la anticuada lista de especificación de requerimientos. Esta lista será detallada en la sección 2.5 del presente documento.

## 2.3 Descripción de objetivos

### 2.3.1 Objetivo General

Desarrollar un servicio web que potencie la participación activa en clases mediante el trabajo colaborativo y en tiempo real, de los códigos presentados en las actividades establecidas. La participación de parte de los alumnos durante las sesiones de clases estará determinada por medio de permisos compartidos, dándole la posibilidad de vista o edición.

### 2.3.2 Objetivo Específico

1. Desarrollar un sistema que contenga un editor de texto que permita la modificación de su contenido en tiempo real y de manera colaborativa.
2. Permitir gestionar sesiones de trabajo académicas entre profesores y estudiantes a través de dinámicas de clases.
3. Definir un administrador de archivos a nivel web que permita gestionar documentos y carpetas a través perfiles de usuario.

## 2.4 Herramientas Utilizadas

Herramienta	Descripción
<b>Angular JS</b>	Es un framework MVC de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA Single-Page Applications.
<b>MongoDB</b>	Es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.
<b>Etherpad</b>	Es un editor web basado en la colaboración en tiempo real.
<b>Socket.io</b>	Es una librería JavaScript para aplicaciones web en tiempo real. Permite la comunicación bidireccional en tiempo real entre los clientes Web y los servidores.
<b>Express</b>	Es una infraestructura de aplicaciones web Node.js mínima y flexible que proporciona un conjunto sólido de características para las aplicaciones web y móviles.
<b>Node.js</b>	Es un entorno Javascript del lado del servidor, basado en eventos.
<b>Mongoose</b>	Es una herramienta de modelado de objetos MongoDB diseñada para funcionar en un entorno asíncronico.

*Tabla 1. Herramientas a utilizar para el desarrollo del proyecto.*

## 2.5 Especificación de Requisitos (Product Backlog)

A continuación se presenta la tabla de los Requisitos funcionales (tabla 2) de la cual consta el proyecto a entregar.

Código de Requisito	Nombre	Prioridad	VALIDADO	Descripción
			(S/N, fecha)	
<b>Sprint 1</b>				
· Autenticación de Perfiles				
RF-1	<b>Login</b>	1	S 09-10-2018	Para permitir el registro y autenticación en el sistema, se utilizarán los siguientes datos: email y contraseña.
<b>Sprint 2</b>				
· Gestión de Sesiones · Primera versión de los editores de texto individual y colaborativo				
RF-2	<b>CRUD de Usuarios</b>	1	S 09-10-2018	El sistema permitirá al administrador crear, ver, modificar y eliminar a los usuarios.
RF-3	<b>Administración de Sesión</b>	1	S 09-10-2018	El sistema permitirá al usuario crear, ingresar o abandonar una sesión. Siendo una sesión, la vista que presenta los dos editores de texto, uno de estos será el que pertenece a la sesión, el cual es compartido, mientras tanto el otro editor pertenecerán a cada usuario por separado.  El usuario que posea los permisos necesarios, podrá realizar la creación de una sesión. Si el usuario que creó la sesión la abandona, ésta procederá a eliminarse junto con la salida de este.
RF-4	<b>Categorizar sesiones</b>	1	S 09-10-2018	El sistema permitirá al usuario, al momento de crear una sesión, diferenciar ésta entre "pública" o "privada". En caso de ser esta una sesión privada, se solicitará una contraseña al momento de unirse a la sesión.
RF-5	<b>Uso de editor de texto individual</b>	1	S 09-10-2018	El sistema permitirá al usuario utilizar un editor de texto individual y único para el usuario.
RF-6	<b>Uso del editor de texto compartido</b>	1	S 09-10-2018	El sistema permitirá al dueño de la sesión, hacer uso de un editor de texto compartido, el cual mostrará en tiempo real el contenido de este para todos aquellos que ingresen a dicha sesión, independiente de ésta sea pública o privada.
RF-7	<b>Guardar contenido del editor de texto</b>	1	S 09-10-2018	El sistema permitirá al usuario: descargar el contenido desarrollado en el editor de texto, y almacenarlo en el servidor.

<b>Sprint 3</b>				
<ul style="list-style-type: none"> <li>· Versión final de los editores de texto individual y colaborativo</li> <li>· Marcado de sintaxis de lenguajes de programación</li> <li>· Manejo de Archivos</li> <li>· Manejo de Carpetas</li> </ul>				
RF-8	<b>Otorgar permisos</b>	1	S 09-10-2018	El creador de la sesión podrá otorgar permisos de escritura al editor de texto compartido entre los participantes de esta sesión.
RF-9	<b>Modificar el editor de texto de la sesión</b>	1	S 09-10-2018	El sistema permitirá al dueño/creador de la sesión el poder modificar distintas funciones del editor de texto principal de la sesión creada, como lo es la sintaxis de algún lenguaje de programación, destacar, entre otros.
RF-10	<b>CRUD de Carpetas</b>	1	S 09-10-2018	El sistema permitirá al usuario crear, ver, modificar y eliminar las carpetas.
RF-11	<b>CRUD de Archivos</b>	1	S 09-10-2018	El sistema además de contar con un CRUD de Carpetas, tendrá la posibilidad de realizar una creación, vista, modificación y eliminación de archivos que pueden o no estar contenidos en una carpeta.
RF-12	<b>Compartir Carpetas y/o Archivos</b>	1	S 09-10-2018	El usuario tendrá la posibilidad de dar permisos de visualización o edición de aquellas carpetas y/o archivos que tenga almacenados dentro del sistema.

*Tabla 2. Requisitos del Proyecto*

## 2.6 Estructura organizacional

### 2.6.1 Roles



Figura 1. Definición de Roles Iniciales

### 2.6.2 Responsabilidades Iniciales

Rol	Descripción	Integrante(s)
Jefe de Proyecto	Representa, lidera y organiza al equipo de trabajo. Realiza un seguimiento periódico del estado del proyecto.	Diego Valencia
Ingeniero de Software	Verifica que el producto cumpla con los requerimientos del cliente, además de enfocarse en dar solución a los problemas planteados.	Diego Valencia
Desarrollador	Personal encargado del análisis e implementación del sistema, utilizando las tecnologías y técnicas establecidas.	Manuel Tapia Daniela Oñate Fabián Urrutia
Diseñador	Responsable de las actividades referentes al diseño gráfico del sistema web a desarrollar (Interfaz de usuario, metáforas, etc).	Fabián Urrutia
Encargado de Documentos	Encargado de la realización de bitácoras, informes y documentos en general.	Luis Lecaros Charlotte Soto Manuel Tapia
Encargado del Cronograma	Encargado de preparar la carta gantt junto con los responsables de los trabajos, actualizar periódicamente el cronograma, fijar los plazos de entrega, etc.	Luis Lecaros
Tester de Software	Encargado de realizar las pruebas al software desarrollado y de informar los resultados de estas. Este rol no tiene mucha relación con el desarrollo del producto para así no comprometer el resultado de las pruebas.	Manuel Tapia Charlotte Soto
Relacionador Público	Es quien interactúa con el cliente, realiza reuniones, identifica los requerimientos del proyecto, establece una buena comunicación e informa al equipo de trabajo.	Daniela Oñate

Tabla 3. Responsabilidades Iniciales

## 2.7 Gestión de riesgos

A continuación en la siguiente tabla (tabla 4) se dará la definición de los riesgos del proyecto, con su probabilidad e impacto

### 2.7.1 Tabla de riesgo del proyecto

Impacto de los riesgos:

1. Catastrófico
2. Crítico
3. Marginal
4. Despreciable

RIESGO	PROBABILIDAD	IMPACTO
Internet deficiente	90%	2
Errores en el sistema web	60%	2
Mal seguimiento de la metodología de desarrollo	50%	2
Herramienta de gestión de proyectos deficiente	49%	4
Fechas de entrega muy ajustadas	45%	2
Escasez de documentación sobre herramientas a utilizar	43%	2
Daño de algún aparato o equipo de desarrollo	40%	2
No desarrollar los sprints completamente en las fechas establecidas.	40%	2
La tecnología utilizada no alcanza las expectativas	39%	3
Documentación del proyecto deficiente.	38%	3
No se aplican estándares establecidos	38%	2
Miembro del equipo realiza de manera deficiente las actividades establecidas	36%	2
Deficiente formación del equipo de trabajo	35%	3
Los costos superan a lo planificado	35%	2
Retraso en entregas al cliente	35%	1
Roles mal establecidos en el equipo de trabajo	28%	3
Ausencia de algún integrante del equipo	20%	3
Miembro se retira del equipo de trabajo.	18%	2

*Tabla 4. Riesgos del Proyecto*

## 2.8 Costeo

En la siguiente tabla (tabla 5), se presenta el presupuesto de costos necesario para que el equipo de trabajo pueda realizar el proyecto a fin.

	Hora presencial	Hora no presencial	Costo unitario / hora	Cant #	Aplicación porcentaje políticas de la Empresa	Subtotales	Totales nivel 1	% del total
<b>1.- Gestión del Proyecto</b>							\$ 21,226,864	94%
Gerente del Proyecto	132	556	\$ 6,980	1		\$ 4,802,240		
Desarrollador	132	556	\$ 4,120	2		\$ 5,669,120		
Encargado de Gestión de Informes y Análisis	132	556	\$ 5,211	3		\$ 10,755,504		
<b>2.- Hardware</b>							\$ 971,993	4%
Computadores			\$ 1,149,990	6	12%	\$ 827,992.80		
Servidores			\$ 1,200,000	1	12%	\$ 144,000		
<b>3.- Software</b>							\$ 2,059	0%
Microsoft Office			\$ 5,720	3	12%	\$ 2,059.20		
Base de Datos (MongoDB +Mongoose)			\$ -			\$ -		
Redmine			\$ -			\$ -		
MEAN Stack (MongoDB, Express, Angular and Node)			\$ -			\$ -		
GitHub			\$ -			\$ -		
Etherpad			\$ -			\$ -		
Discord			\$ -			\$ -		
<b>4.- Otros</b>							\$ 276,233	1%
Alimentación [3]			\$ 420,970	3 meses	12%	\$ 50,516		
Instalación Unidad de Producción			\$ 1,670,000	3 meses	12%	\$ 200,400		
Telefonía + Internet 100 mb [2]			\$ 143,970	3 meses	12%	\$ 17,276		
Insumos generales (luz, agua)			\$ 67,000	3 meses	12%	\$ 8,040		
<b>Total Costo Estimado del Proyecto</b>							\$ 22,477,149	100%

Tabla 5. Costos asociados al proyecto.

## 2.9 Avance por Sprint

### 2.9.1 Sprint 1

#### 2.9.1.1 Sprint Backlog

A continuación en la siguiente tabla (tabla 6) se presentarán los diversos requisitos que fueron cumplidos en la realización de este Sprint.

Código de Requisito	Nombre	Prioridad	VALIDADO	Descripción
			(S/N, fecha)	
<i>Sprint 1</i>				
· Autenticación de Perfiles				
RF-1	<b>Login</b>	1	S 09-10-2018	Para permitir el registro y autenticación en el sistema, se utilizarán los siguientes datos: email y contraseña.

Tabla 6. Requisitos definidos para el Primer Sprint

## 2.9.1.2 Diseño

### 2.9.1.2.1 Diagrama de Caso de uso

En el siguiente diagrama (Diagrama 1) se presenta el Diagrama de Caso de uso correspondiente al Sprint 1, dando la posibilidad a un Usuario que ya tenga una cuenta existente con anterioridad poder Loguearse al sistema o bien, dar la posibilidad a un nuevo Usuario de poder crear su cuenta dentro del sistema accediendo a las funcionalidades que este presentará.

*Diagrama 1. Diagrama de Caso de uso Sprint 1*

### 2.9.1.2.2 Descripción de Casos de uso

<b>Nombre</b>	Login
<b>Descripción</b>	El usuario ingresa los datos solicitados por el sistema para iniciar sesión.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario selecciona la opción de “Login” para ingresar al sistema.
2	El sistema solicita el correo y la contraseña del usuario que desea ingresar.
3	El usuario ingresa los datos solicitados.
4	El sistema verifica los datos ingresados.
5	Si los datos son correctos el sistema procede a ingresar a la pantalla principal y finaliza el caso de uso.
<b>Excepciones</b>	

Paso 5	Acción
E.5.1	Si los datos del usuario son incorrectos el sistema manda un aviso a pantalla: “Los datos ingresados son incorrectos”.
E.5.2	El usuario puede volver a ingresar los datos.

*Tabla 7. Descripción del caso de uso “Login”.*

<b>Nombre</b>	Crear Usuario
<b>Descripción</b>	Se ingresan los datos necesarios para poder registrar un nuevo usuario al sistema.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	Se selecciona la opción “Crear Usuario”.
2	El sistema solicita los siguientes datos de la persona que se desea registrar: correo, nombre completo y contraseña.
3	El usuario ingresa los datos solicitados.
4	El sistema verifica que el correo ingresado no se esté utilizando actualmente por otro usuario ya registrado.
5	Si el correo no es utilizado por otra persona ya registrada, se procede a registrar al nuevo usuario y se finaliza el caso de uso.
<b>Excepciones</b>	
<b>Paso 5</b>	<b>Acción</b>
E.5.1	Si el correo es utilizado por otra persona ya registrada el sistema manda un aviso a pantalla: “El correo ingresado ya está registrado”.
E.5.2	El usuario puede volver a ingresar los datos para registrarse.

*Tabla 8. Descripción del caso de uso “Crear Usuario”.*

2.9.1.2.3 Diagrama de Secuencia

En los siguientes diagramas (Diagramas 2, 3 ) se presenta Diagramas de Secuencia correspondiente al Sprint 1.

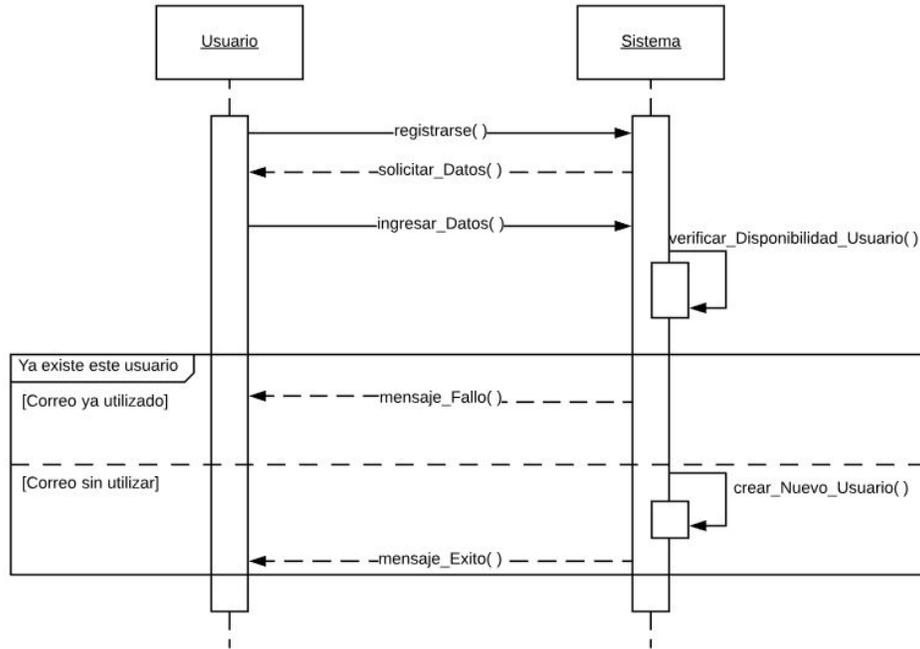


Diagrama 2. Diagrama de Secuencias “Registrar Usuario”

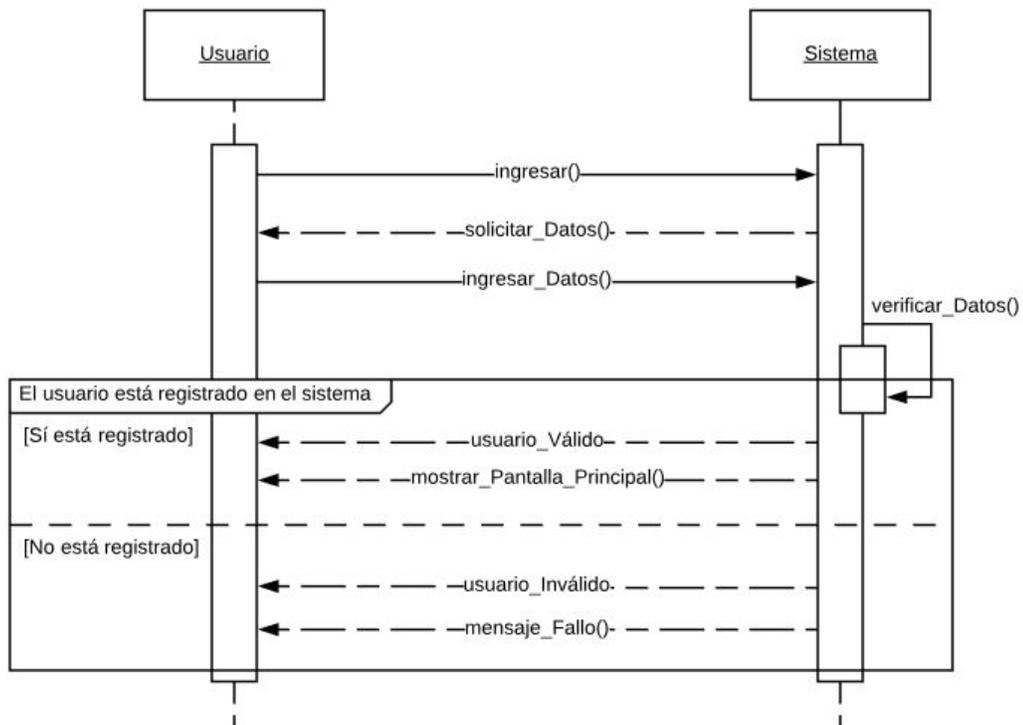


Diagrama 3. Diagrama de Secuencias “Iniciar Sesión”

#### 2.9.1.2.4 Diagrama de Clases

En el siguiente diagrama (Diagrama 4) se presenta el Diagrama de Clases correspondiente al Sprint 1. En este sprint solamente se ha implementado la clase “Usuario”, el cual presentará los atributos: correo con el cual se registro el usuario, nombre completo para ámbitos de administración y la contraseña para ingresar al sistema.

Diagrama 4. Diagrama de Clases Sprint 1

#### 2.9.1.2.5 Modelado de Base de Datos

A continuación se presenta el diagrama (Diagrama 5) del modelado de la base de datos para el proyecto, concebido desde el Sprint 1.

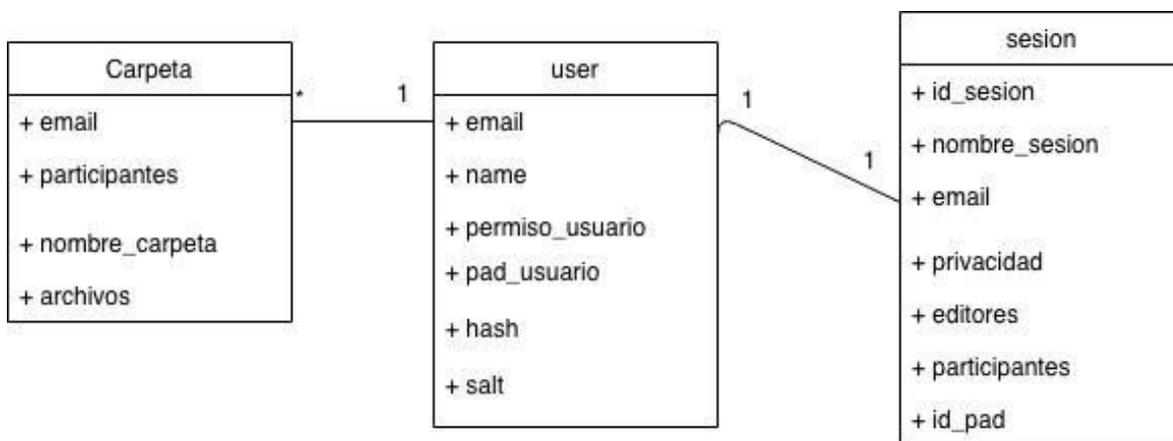
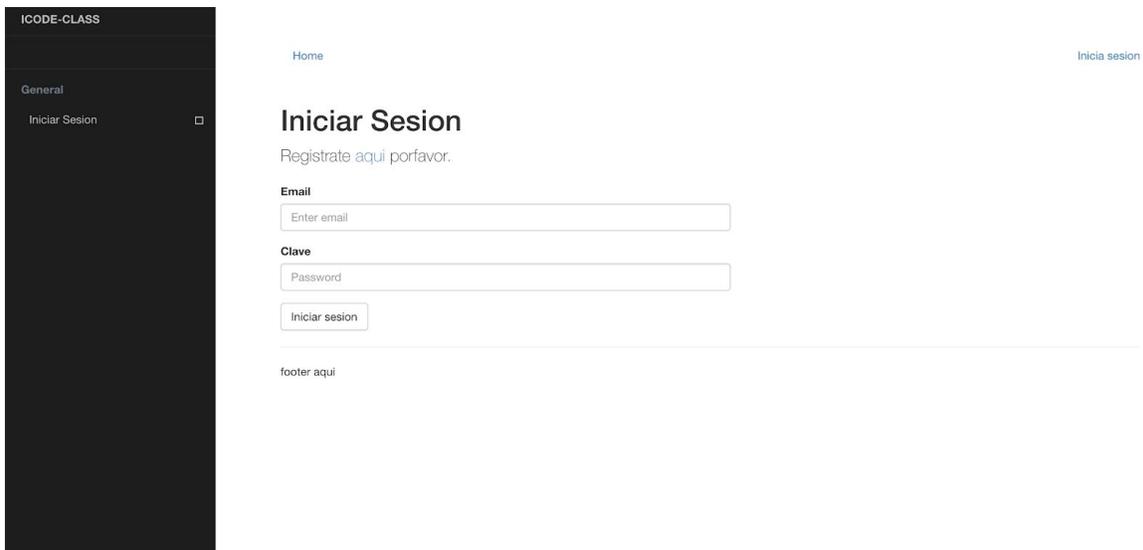


Diagrama 5. Diagrama de Modelo Base de Datos Sprint 1

### 2.9.1.3 Implementación (Vistas)

A continuación se muestra en la siguiente ilustración (Ilustración [2 - 4]) las respectivas vistas del sistema web o producto.

*Ilustración 2. Vista de Inicio preliminar.*



*Ilustración 3. Vista de Login preliminar.*

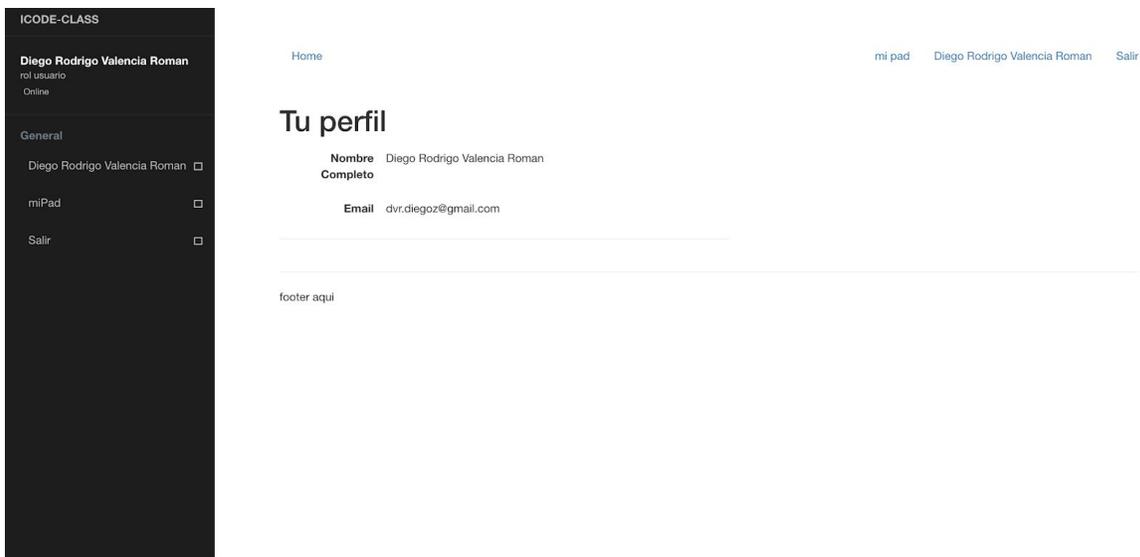


Ilustración 4. Vista de Perfil preliminar.

### 2.9.1.4 Aspectos Generales

#### 2.9.1.4.1 Responsables por actividad

Sprint 1	
Actividad	Responsable
Desarrollar un sistema de autenticación de una serie de perfiles	Daniela Oñate Fabián Urrutia Diego Valencia
Definir los perfiles de usuario del sistema	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Diseñar el sistema	Daniela Oñate Fabián Urrutia Diego Valencia
Diseñar el sistema de registro de usuario	Daniela Oñate Fabián Urrutia Diego Valencia
Diseñar el sistema de autenticación de usuario	Daniela Oñate Fabián Urrutia Diego Valencia
Desarrollar e implementar los módulos del sistema	Daniela Oñate Fabián Urrutia Diego Valencia
Desarrollar e implementar el módulo de registro en el servidor del sistema web	Daniela Oñate

	Fabián Urrutia Diego Valencia
Desarrollar e implementar el módulo de autenticación en el servidor del sistema web	Daniela Oñate Fabián Urrutia Diego Valencia
Diseñar e implementar las diversas vistas del sistema	Daniela Oñate Fabián Urrutia Diego Valencia
Diseñar e implementar la vista del registro del sistema web	Daniela Oñate Fabián Urrutia Diego Valencia
Diseñar e implementar la vista de la autenticación del sistema web	Daniela Oñate Fabián Urrutia Diego Valencia
Modelar base de datos global	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Definición de requisitos	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Reunión con el cliente	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Educción de requisitos	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Presentación del 1er sprint al cliente	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia

*Tabla 9. Responsabilidades actualizadas para el Sprint 1*

### 2.9.1.4.2 Avance según Carta Gantt

En la siguiente figura (figura 2) se detalla el avance realizado según Carta Gantt para el Sprint 1 fue de un 100%, realizando todas las actividades que fueron planificadas para este Sprint, destacando la educación de requisitos y la presentación del primer módulo del producto al cliente, el cual estaba enfocado meramente en la autenticación de usuarios.



Figura 2. Avance de Carta Gantt hasta final de Primer Sprint

## 2.9.2 Sprint 2

### 2.9.2.1 Sprint Backlog

A continuación se presentarán los diversos requisitos que fueron definidos para la realización de este Sprint.

Código de Requisito	Nombre	Prioridad	VALIDADO	Descripción
			(S/N, fecha)	
<b>Sprint 2</b> <ul style="list-style-type: none"> <li>Gestión de Sesiones</li> <li>Primera versión de los editores de texto individual y colaborativo</li> </ul>				
RF-2	<b>Administración de Sesión</b>	1	S 09-10-2018	<p>El sistema permitirá al usuario crear, ingresar o abandonar una sesión. Siendo una sesión, la vista que presenta los dos editores de texto, uno de estos será el que pertenece a la sesión, el cual es compartido, mientras tanto el otro editor perteneciera a cada usuario por separado.</p> <p>El usuario que posea los permisos necesarios, podrá realizar la creación de una sesión. Si el usuario que creó la sesión la abandona, ésta procederá a eliminarse junto con la salida de este.</p>
RF-3	<b>Categorizar sesiones</b>	1	S 09-10-2018	El sistema permitirá al usuario, al momento de crear una sesión, diferenciar ésta entre "pública" o "privada". En caso de ser esta una sesión privada, se solicitará una contraseña al momento de unirse a la sesión.
RF-4	<b>Uso de editor de texto individual</b>	1	S 09-10-2018	El sistema permitirá al usuario utilizar un editor de texto individual y único para el usuario.
RF-5	<b>Uso del editor de texto compartido</b>	1	S 09-10-2018	El sistema permitirá al dueño de la sesión, hacer uso de un editor de texto compartido, el cual mostrará en tiempo real el contenido de este para todos aquellos que ingresen a dicha sesión, independiente de ésta sea pública o privada.
RF-6	<b>Otorgar permisos</b>	1	S 09-10-2018	El creador de la sesión podrá otorgar permisos de escritura al editor de texto compartido entre los participantes de esta sesión.
RF-7	<b>Modificar el editor de texto de la sesión</b>	1	S 09-10-2018	El sistema permitirá al dueño/creador de la sesión el poder modificar distintas funciones del editor de texto principal de la sesión creada, como lo es la sintaxis de algún lenguaje de programación, destacar, entre otros.

*Tabla 10. Requisitos definidos para el Segundo Sprint*

## 2.9.2.2 Diseño

### 2.9.2.2.1 Diagrama de Caso de uso

En el siguiente diagrama (Diagrama 6) se presenta el Diagrama de Caso de uso correspondiente al Sprint 2.

*Diagrama 6. Diagrama de Caso de uso Sprint 2*

## 2.9.2.2.2 Descripción de Casos de uso

<b>Nombre</b>	Crear sesión
<b>Descripción</b>	El usuario crea una sesión con los parámetros elegidos.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario selecciona la opción “Crear sesión” para crear una sesión.
2	El sistema le solicita al usuario el identificador de la sesión (“id_sesion”, el cuál debe ser único), el nombre de la sesión a crear y la privacidad.
3	El usuario ingresa los datos solicitados.
4	El sistema verifica que el identificador de la sesión (“id_sesion”) no se esté utilizando actualmente.
5	Si el “id_sesion” no está siendo utilizado, el sistema procede a crear la sesión con los datos ingresados y a mostrar el ambiente de esta en pantalla. Finaliza el caso de uso.
<b>Excepciones</b>	
<b>Paso 5</b>	<b>Acción</b>
E.5.1	Si el “id_sesion” ya está siendo utilizado, el sistema manda un mensaje a pantalla “El identificador de la sesión ya está siendo utilizado”.
E.5.2	El usuario puede volver a ingresar el identificador de la sesión.

Tabla 11. Descripción del caso de uso “Crear Sesión”.

<b>Nombre</b>	Unirse a la sesión
<b>Descripción</b>	El usuario ingresa a una sesión ya creada con anterioridad.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario selecciona la opción de “Unirse a la sesión” para ingresar a una sesión ya creada.
2	El sistema le concede el acceso al usuario, otorgándole los permisos que posee con anterioridad.
<b>Excepciones</b>	
<b>Paso 2</b>	<b>Acción</b>
E.2.1	El sistema niega el acceso al usuario debido a que la sesión posee un tipo de privacidad que le bloquea el acceso a cualquier usuario.
E.2.2	El usuario ingresa “id_sesión” con la cual verifica que posee permiso para poder ingresar
E.2.3	El sistema le concede el acceso al usuario, otorgándole los permisos que posee con anterioridad.

*Tabla 12. Descripción del caso de uso “Unirse a la sesión”.*

<b>Nombre</b>	Otorgar permisos
<b>Descripción</b>	El dueño de la sesión le otorga permisos de edición del editor general a un usuario que esté dentro de esta.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El dueño de la sesión selecciona la opción “Otorgar Permisos” a uno de los usuarios listados que esté dentro de la sesión.
2	El sistema verifica si el correo del usuario ya está dentro del listado de editores de la sesión, de no ser así, modifica el permiso que posee el usuario y añade su correo al listado de editores de la sesión.
<b>Excepciones</b>	
<b>Paso 5</b>	<b>Acción</b>
E.2.1	El sistema determina que el usuario ya posee permisos de edición por lo que envía un mensaje al dueño de sesión informando esta situación.

*Tabla 13. Descripción del caso de uso “Otorgar permisos”.*

<b>Nombre</b>	Quitar permisos
<b>Descripción</b>	El administrador de la sesión puede quitar los permisos que ha otorgado a los usuarios dentro de su sesión. El usuario a quién se le quiten los permisos ya no podrá utilizar el editor general de la sesión.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El administrador de la sesión escoge al usuario a quién le quitará los permisos y selecciona la opción “Quitar permisos”
2	El sistema procede a quitar los permisos de utilización del editor general de la sesión al usuario seleccionado.
3	Una vez retirados los permisos, el sistema manda un mensaje “Se le han quitado los permisos al usuario: nombreusuario”. Se finaliza el caso de uso.

*Tabla 14. Descripción del caso de uso “Quitar permisos”.*

<b>Nombre</b>	Finalizar sesión
<b>Descripción</b>	El administrador de la sesión realiza la finalización de esta.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El administración de la sesión selecciona la opción “Finalizar sesión”.
2	El sistema procede a cerrar y eliminar la sesión y los datos respectivos de la misma. Cabe destacar que una vez eliminada la sesión, el id utilizado para esta ahora estará habilitado para otra sesión.
3	Una vez eliminada la sesión el sistema manda un mensaje “La sesión ha sido eliminada”.
<b>Excepciones</b>	
<b>Paso</b>	<b>Acción</b>

*Tabla 15. Descripción del caso de uso “Finalizar sesión”.*

<b>Nombre</b>	Modificar editor de texto de la sesión
<b>Descripción</b>	El usuario al estar en una sesión propia puede modificar la sintaxis del lenguaje de programación utilizado en el editor de texto y la tipografía.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario selecciona la opción “Configuraciones del editor de texto” para modificar características del editor de texto.
2	El sistema muestra las opciones de “Cambiar lenguaje de programación” y de “Cambiar tipografía”.
3	El usuario selecciona la opción “Cambiar lenguaje de programación” para cambiar la sintaxis del editor de texto.
4	El sistema muestra un menú en pantalla todas las sintaxis de los lenguajes de programación que se pueden usar.

5	El usuario selecciona la sintaxis que desea ocupar.
6	El sistema cambia la sintaxis del editor de texto por la que el usuario ha seleccionado.
7	El editor de texto se carga nuevamente con la sintaxis elegida. Finaliza el caso de uso.
<b>Excepciones</b>	
<b>Paso 5</b>	<b>Acción</b>
E.3.1	El usuario selecciona la opción “Cambiar tipografía” para cambiar el tipo de letra utilizado en el editor de texto.
E.3.2	El sistema muestra un menú en pantalla de las tipografías que se pueden utilizar en el editor de texto.
E.3.3	El usuario selecciona la tipografía que desea ocupar.
E.3.4	El sistema cambia la tipografía del editor de texto por la que el usuario ha seleccionado.
E.3.5	El editor de texto se carga nuevamente con la tipografía elegida. Finaliza el caso de uso.

*Tabla 16. Descripción del caso de uso “Modifica editor de texto de la sesión”.*

2.9.2.2.3 Diagrama de Secuencia

En los siguientes diagramas (Diagrama 7, 8 , 9, 10, 11 y 12) se presentarán los Diagramas de Secuencia correspondiente al Sprint 2.

En el siguiente diagrama (Diagrama 7) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Crear Sesión”.

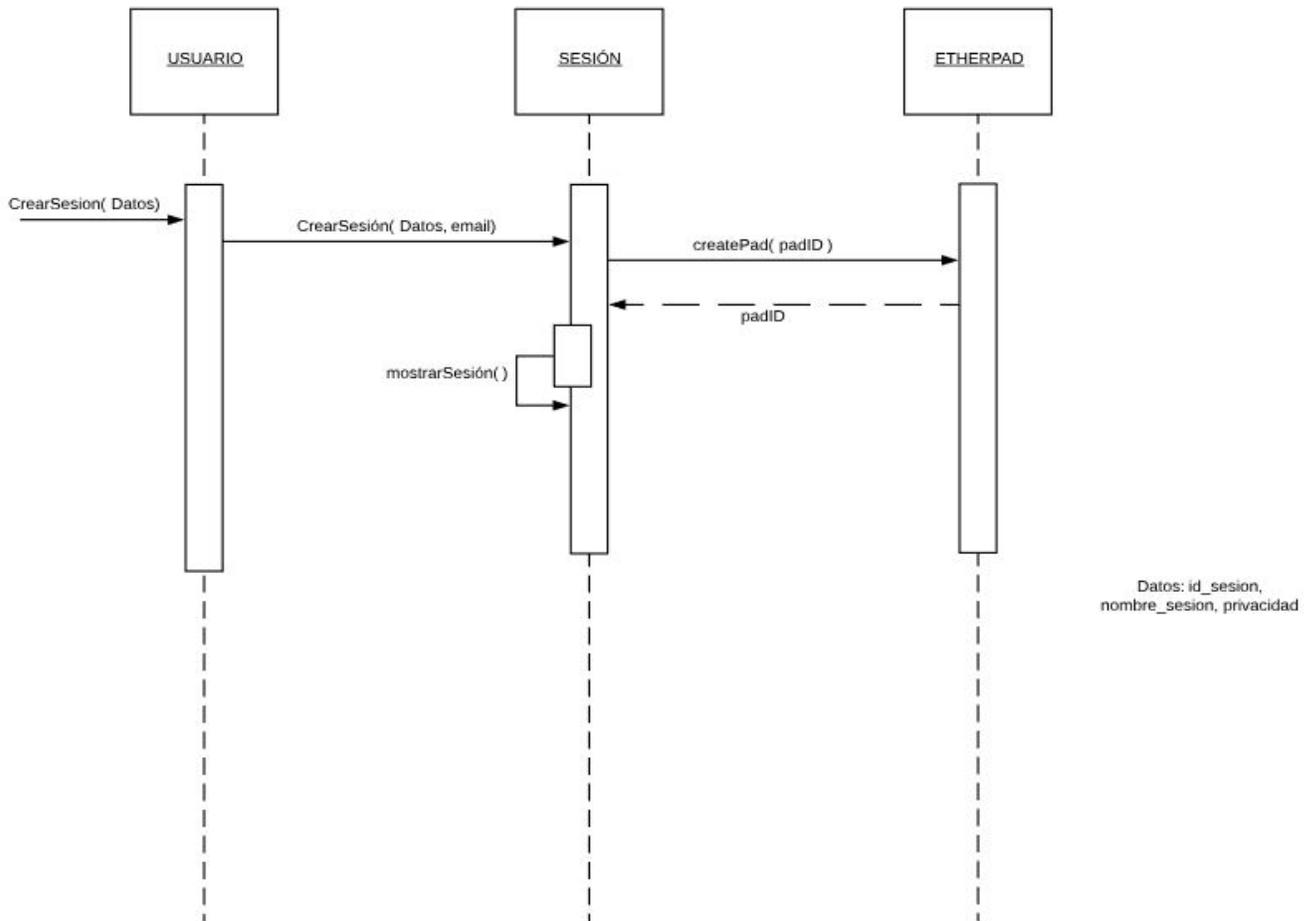


Diagrama 7. Diagrama de Secuencia “Crear Sesión”

En el siguiente diagrama (Diagrama 8) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Unirse a Sesión”.

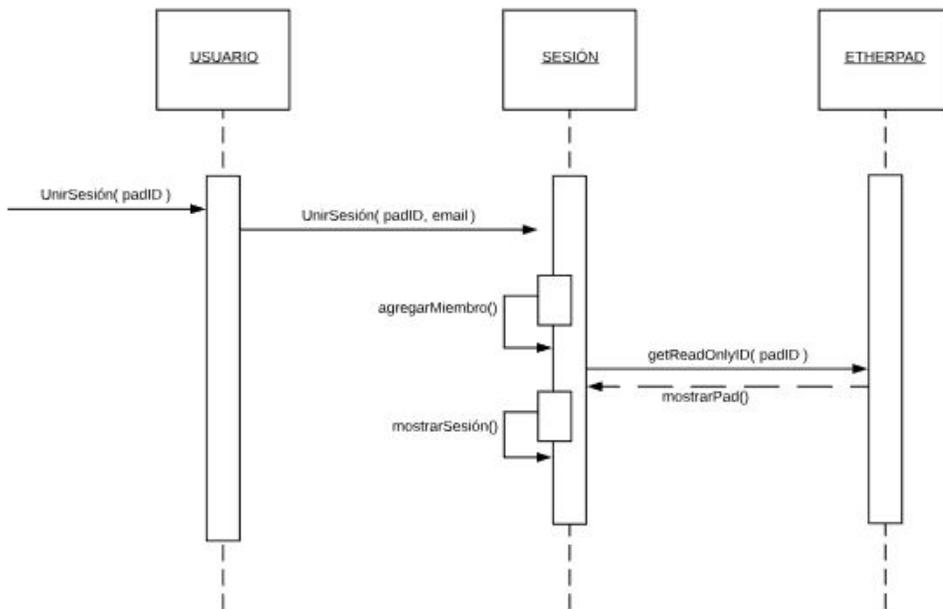


Diagrama 8. Diagrama de Secuencia “Unirse a Sesión”

En el siguiente diagrama (Diagrama 9) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Otorgar Permisos”.

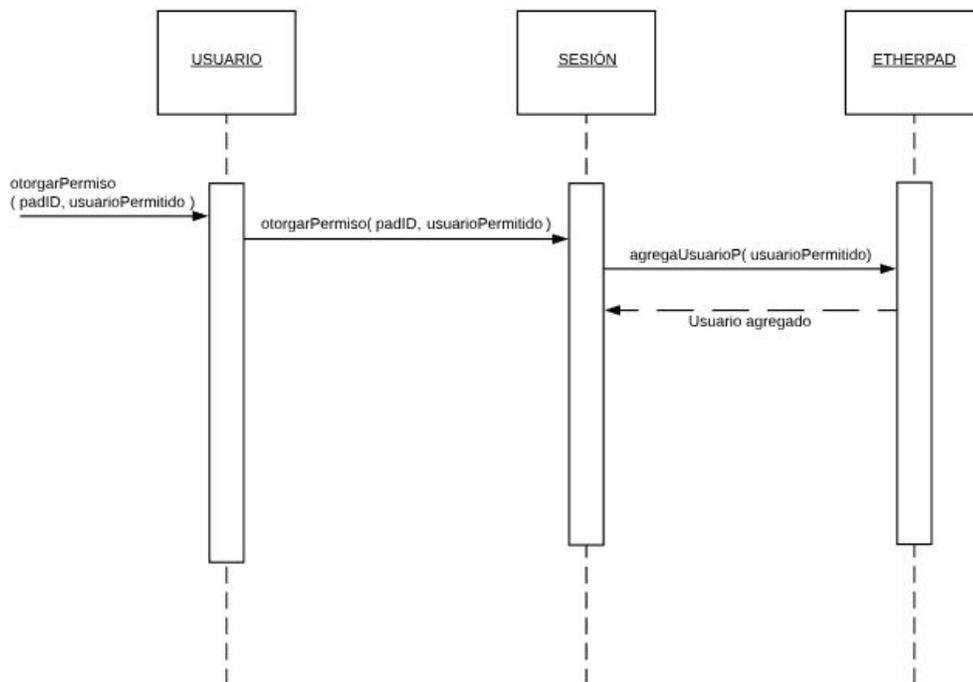


Diagrama 9. Diagrama de Secuencia “Otorgar Permisos”

En el siguiente diagrama (Diagrama 10) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Quitar Permisos”.

*Diagrama 10. Diagrama de Secuencia “Quitar Permisos”*

En el siguiente diagrama (Diagrama 11) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Finalizar sesión”.

*Diagrama 11. Diagrama de Secuencia “Finalizar Sesión”*

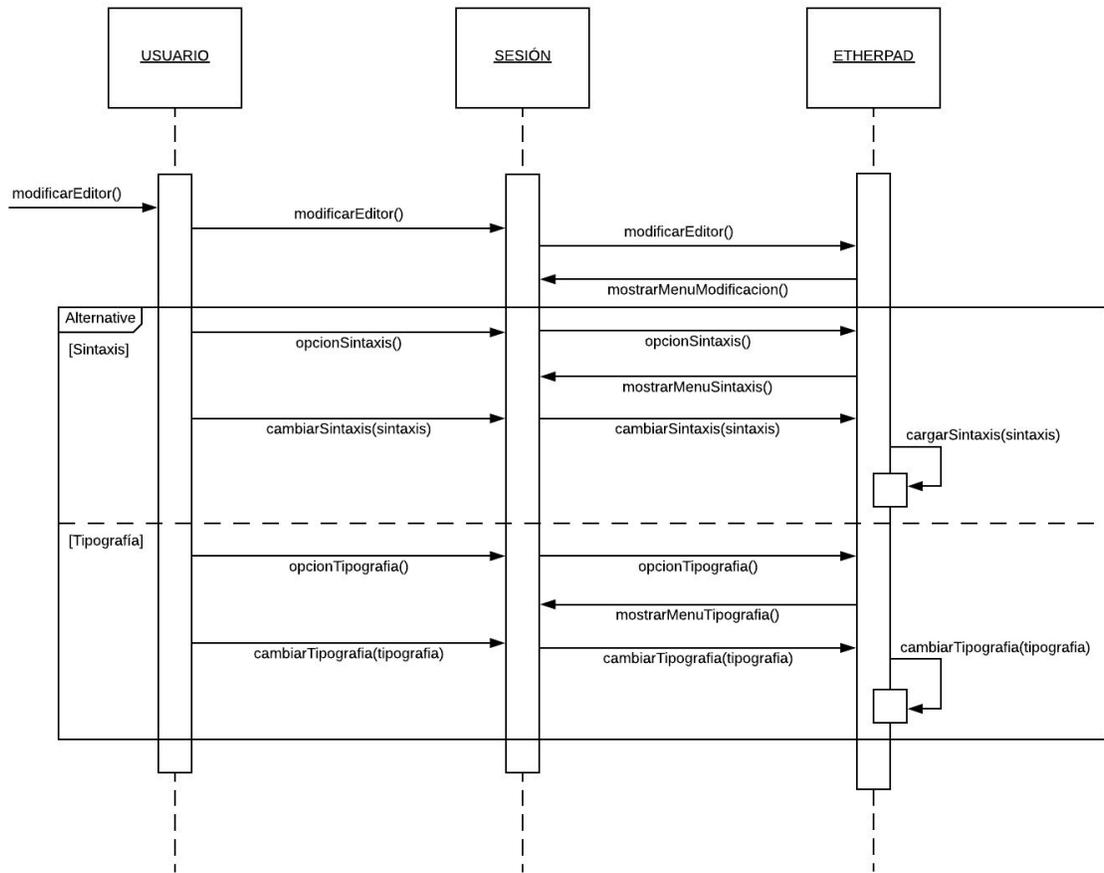


Diagrama 12. Diagrama de Secuencia “Modificar editor de texto de la sesión”

2.9.2.2.4 Diagrama de Clases

A continuación se presenta el diagrama de clases (diagrama 12) concebido desde el Sprint 1, modificándose para el Sprint 2, se añadieron las clases de Sesión, la cual contiene todos los parámetros que contendrá una sesión al ser creada, destacando el id\_sesión el cual será utilizado para las sesiones privadas y editores, el cual contendrá a todos los usuarios que tienen privilegio de utilizar el editor de texto de la sesión. Por otra parte se añadió la clase Carpeta, la cual contendrá la configuración de una carpeta asociada a un usuario, la cual tendrá en su interior diversos archivos y podrá ser compartida con otros usuarios para acceder a estos archivos.

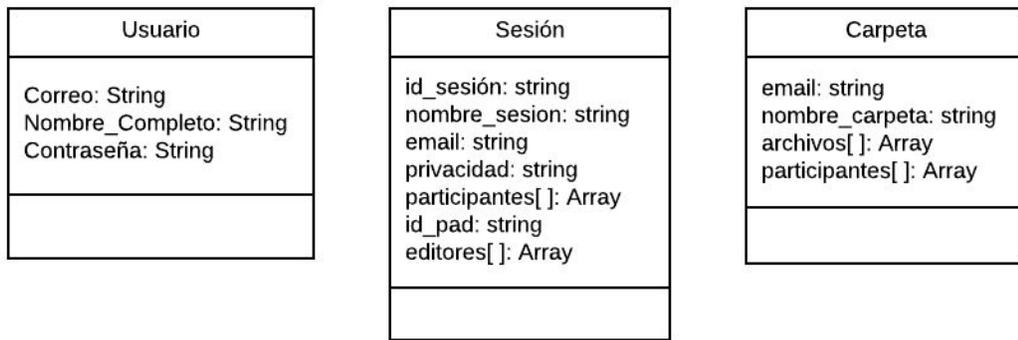


Diagrama 13. Diagrama de clases Sprint 2

2.8.2.2.5 Modelado de Base de Datos

A continuación se presenta el diagrama (Diagrama 13) del modelado de la base de datos para el proyecto, concebido desde el Sprint 1 y manteniéndose en el Sprint 2.

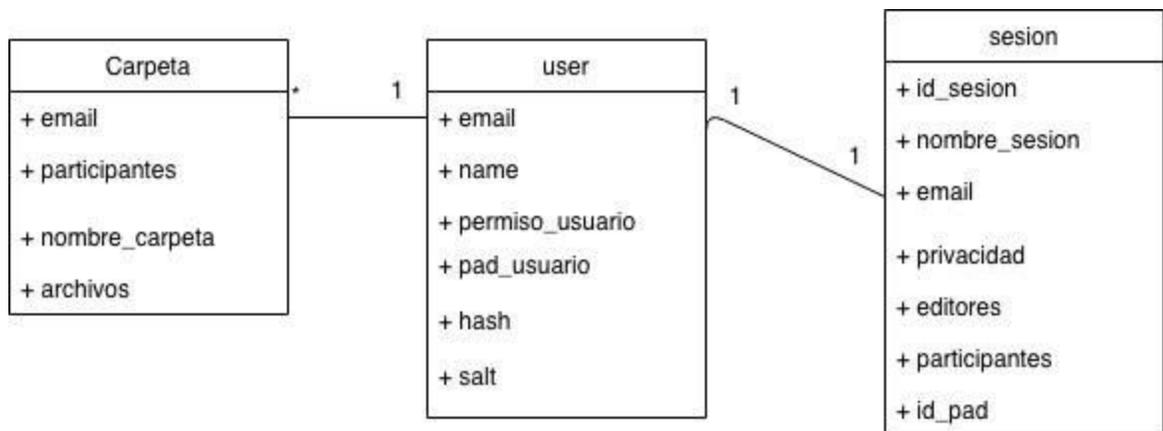


Diagrama 14. Diagrama de Modelo Base de Datos Sprint 2

### 2.9.2.3 Implementación

A continuación se presenta en las siguientes ilustraciones (Ilustración 5,6) un avance preliminar de las vistas relacionadas al Sprint 2, es decir, relacionada a las Sesiones de trabajo.

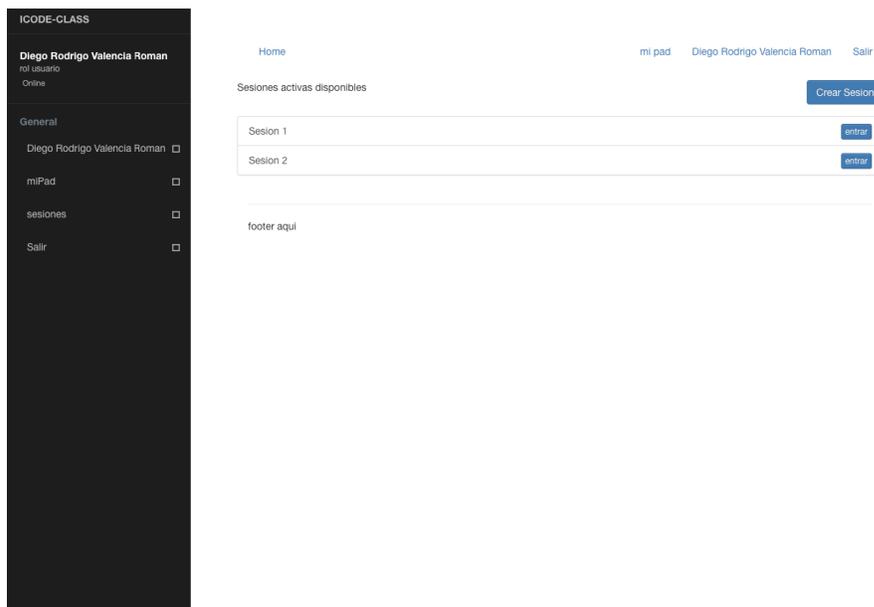


Ilustración 5. Vista preliminar de la pantalla de Sesiones

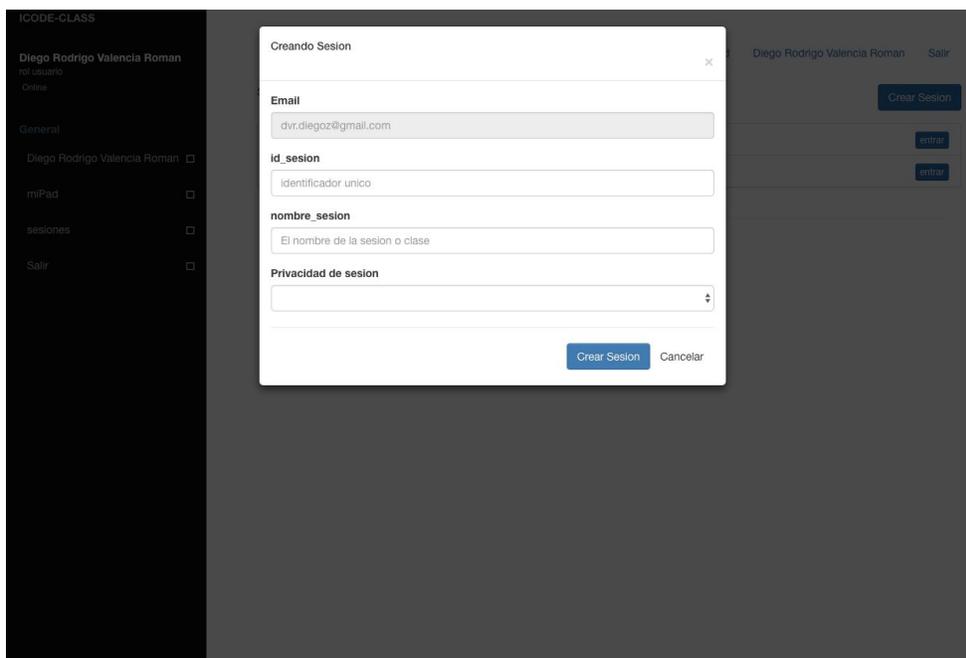


Ilustración 6. Vista preliminar de la pantalla de Creación de Sesiones

## 2.9.2.4 Aspectos Generales

### 2.9.2.4.1 Responsables por actividad

En la siguiente tabla (tabla 17) se describen las actividades contempladas para el Sprint 2, con sus respectivos responsables.

Sprint 2	
Actividad	Responsable
Corrección de errores detectados en presentación	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Desarrollar un sistema de sesiones, en las cuales puedan ingresar los usuarios del sistema	Daniela Oñate Manuel Tapia Diego Valencia
Definir funcionamiento de sesiones	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Definir los tipos de sesiones	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Implementar base de datos enfocada a las sesiones	Daniela Oñate Manuel Tapia Diego Valencia
Definir los usuario que participarán en cada sesión	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Definir los privilegios y restricciones de cada sesión	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Desarrollar e implementar el módulo de sesiones en el servidor del sistema web	Daniela Oñate Manuel Tapia

	Diego Valencia
Diseñar e implementar las vistas de sesiones del sistema web	Daniela Oñate Manuel Tapia Diego Valencia
Desarrollar la versión preliminar del editor colaborativo	Daniela Oñate Manuel Tapia Diego Valencia
Definir el funcionamiento y estructura del editor de texto colaborativo	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Definir las restricciones del editor de texto colaborativo	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Desarrollar e implementar un módulo preliminar de enlace del editor con el sistema web	Daniela Oñate Manuel Tapia Diego Valencia
Presentación del 2do sprint al cliente	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Presentación avance iCode-Class	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia

*Tabla 17. Responsabilidades actualizadas para el Sprint 2*

2.9.2.4.2 Roles actualizados

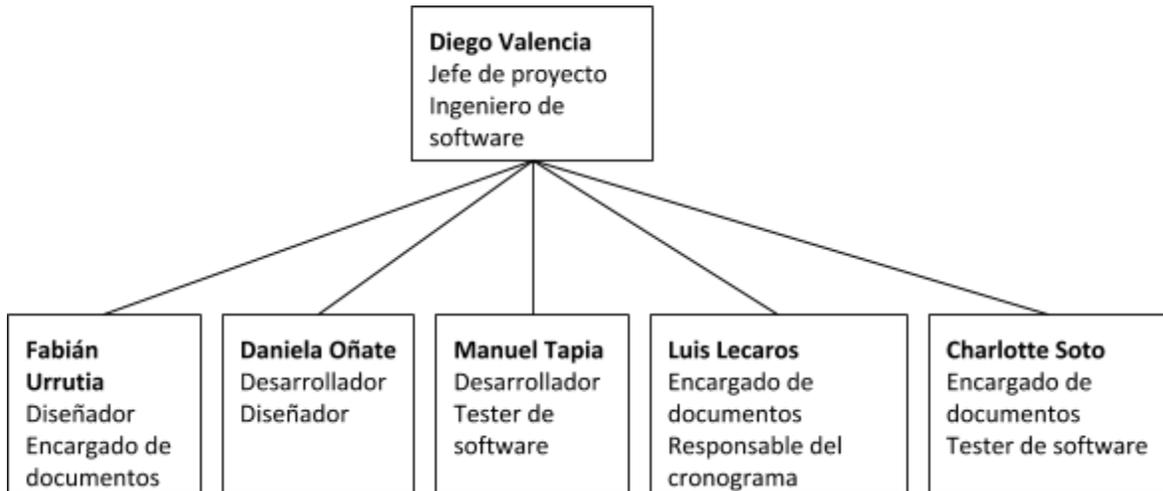


Figura 3. Roles Actualizados para el Segundo Sprint

2.9.2.4.3 Avance según Carta Gantt

En la siguiente figura (figura 4) se presentará el avance realizado según Carta Gantt para el Sprint 2 ha sido de un 50% hasta la fecha de hoy, completando todas las actividades que fueron planificadas para realizarse.

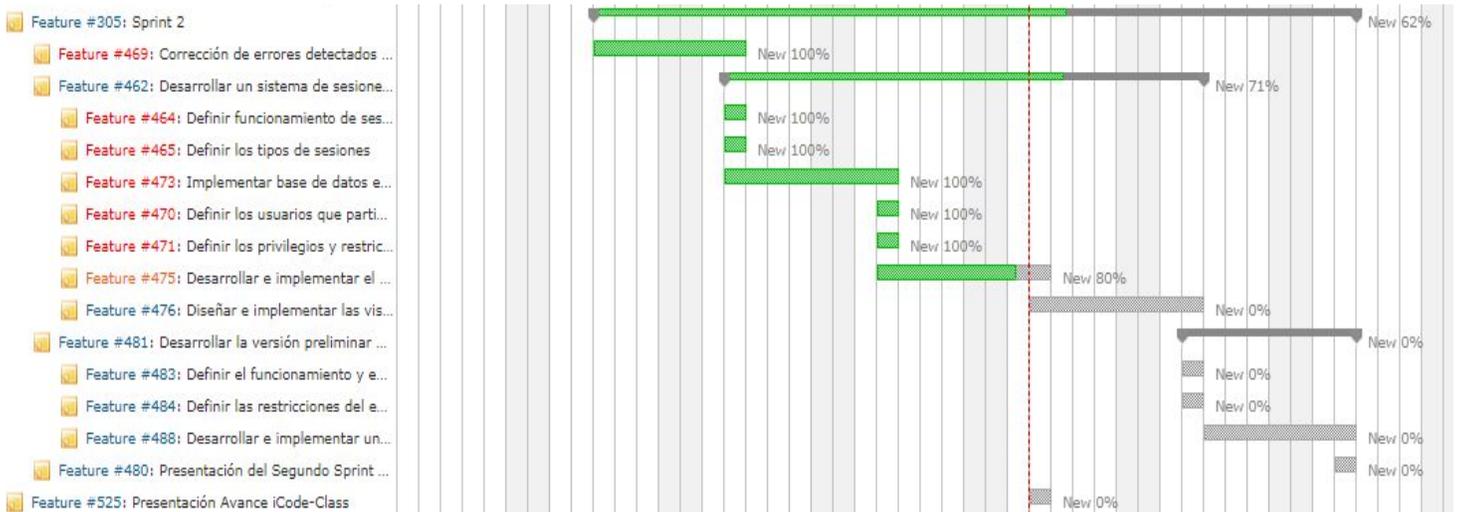


Figura 4. Avance de Carta Gantt hasta mitad de Segundo Sprint

## 2.9.3 Sprint 3

### 2.9.3.1 Sprint Backlog

A continuación se presentarán los diversos requisitos que fueron definidos para la realización de este Sprint.

Código de Requisito	Nombre	Prioridad	VALIDADO	Descripción
			(S/N, fecha)	
<b>Sprint 3</b> <ul style="list-style-type: none"> <li>· Gestión de Archivos</li> <li>· Gestión de Carpetas</li> <li>· Versión final de los editores de texto individual y colaborativo, con todas sus funcionalidades</li> </ul>				
RF-8	<b>CRUD de Usuarios</b>	1	S 11-12-2018	El sistema permitirá al administrador crear, ver, modificar y eliminar a los usuarios.
RF-9	<b>CRUD de Carpetas</b>	1	S 11-12-2018	El sistema permitirá al usuario crear, ver, modificar y eliminar las carpetas.
RF-10	<b>CRUD de Archivos</b>	1	S 11-12-2018	El sistema además de contar con un CRUD de Carpetas, tendrá la posibilidad de realizar una creación, vista, modificación y eliminación de archivos que pueden o no estar contenidos en una carpeta.
RF-11	<b>Compartir Carpetas y/o Archivos</b>	1	S 11-12-2018	El usuario tendrá la posibilidad de dar permisos de visualización o edición de aquellas carpetas y/o archivos que tenga almacenados dentro del sistema.
RF-12	<b>Guardar contenido del editor de texto</b>	1	S 11-12-2018	El sistema permitirá al usuario: descargar el contenido desarrollado en el editor de texto, y almacenarlo en el servidor.

*Tabla 18. Requisitos definidos para el Tercer Sprint*

### 2.9.3.2 Diseño

#### 2.9.3.2.1 Diagrama de Caso de uso

En el siguiente diagrama (Diagrama 15) se presenta el Diagrama de Caso de uso correspondiente al Sprint 3.

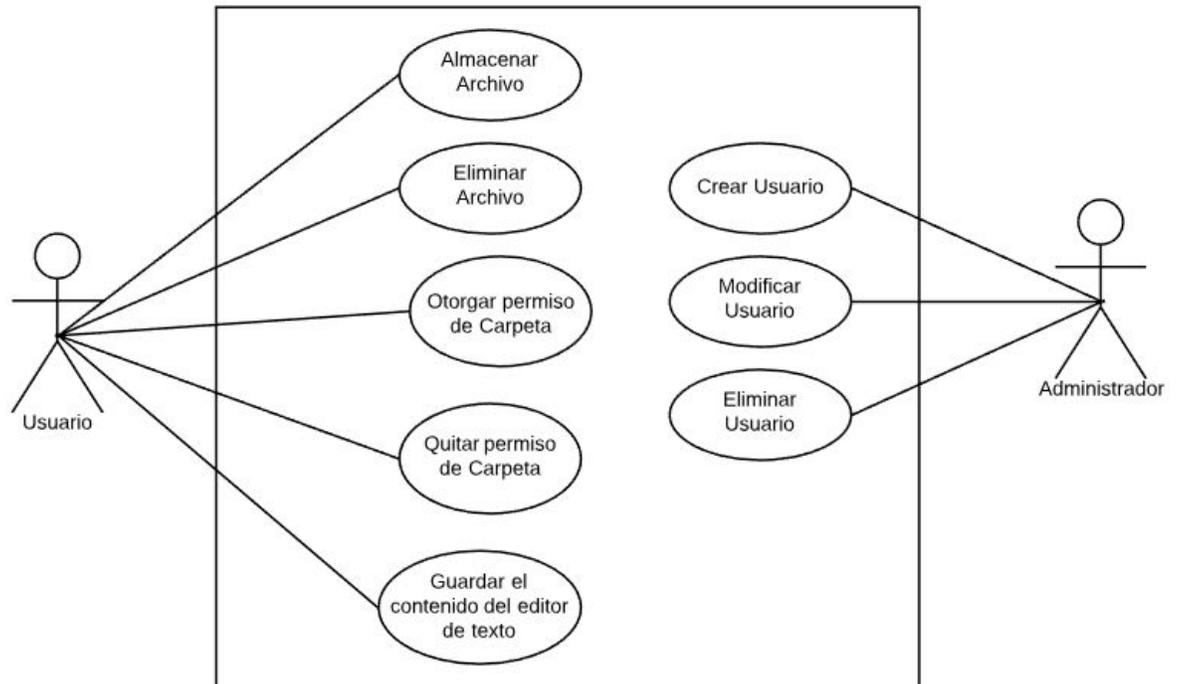


Diagrama 15. Diagrama de Caso de uso Sprint 3

## 2.9.3.2.2 Descripción de Casos de uso

<b>Nombre</b>	Almacenar Archivo
<b>Descripción</b>	El usuario almacena un archivo en la carpeta de su propiedad.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario debe de estar en la vista “Mi Carpeta” y elegir la opción de “Subir”
2	El sistema presentará una ventana emergente de tal manera que el usuario pueda seleccionar el archivo (sólo de texto).
3	El usuario selecciona el archivo a subir a la carpeta.
4	El sistema almacena el archivo en la carpeta, presentando una vista previa del nuevo archivo añadido.

Tabla 19. Descripción del caso de uso “Almacenar Archivo”.

<b>Nombre</b>	Eliminar Archivo
<b>Descripción</b>	El usuario elimina un archivo en la carpeta de su propiedad.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario debe de estar en la vista “Mi Carpeta” y seleccionar su carpeta.
2	El sistema presentará una vista de los archivos contenidos en esa carpeta, para la selección del usuario.
3	El usuario selecciona el archivo que desea utilizar.
4	El sistema presentará una ventana emergente que mostrará el contenido del archivo, como así las siguientes opciones: <ul style="list-style-type: none"> <li>• Bajar</li> <li>• Modificar</li> <li>• Eliminar</li> </ul>
5	El usuario selecciona la opción de “Eliminar”.
6	El Sistema actualiza la vista de la carpeta, ya sin el archivo eliminado.

Tabla 20. Descripción del caso de uso “Eliminar Archivo”.

<b>Nombre</b>	Otorgar permiso de Carpeta
<b>Descripción</b>	El usuario otorga permiso para que un usuario puede hacer uso(sin modificar) del contenido total de su carpeta.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario debe de estar en la vista “Mi Carpeta” y seleccionar en el apartado de “Permiso en tu Carpeta”
2	El sistema presentará un editor de texto para que se haga ingreso del email del usuario al cual desea darle permiso, además de una lista de los emails con permisos y un botón de eliminar en un costado de cada uno.
3	El usuario ingresa el email de otro usuario.
4	El sistema registra ese usuario en el apartado de permisos, y actualiza la lista de los emails con permiso.

*Tabla 21. Descripción del caso de uso “Otorgar Permiso Carpeta”.*

<b>Nombre</b>	Quitar permiso de Carpeta
<b>Descripción</b>	El usuario quita permiso para que un usuario puede hacer uso(sin modificar) del contenido total de su carpeta.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario debe de estar en la vista “Mi Carpeta” y seleccionar en el apartado de “Permiso en tu Carpeta”
2	El sistema presentará un editor de texto para que se haga ingreso del email del usuario al cual desea darle permiso, además de una lista de los emails con permisos y un botón de “eliminar” en un costado de cada uno.
3	El usuario seleccionará el email del usuario al cual desea quitarle el permiso, a través del botón “eliminar” correspondiente.
4	El sistema actualizará el listado de los emails con permisos, sin el email eliminado anteriormente.

*Tabla 22. Descripción del caso de uso “Quitar Permiso Carpeta”.*

<b>Nombre</b>	Guardar el contenido del editor de texto
<b>Descripción</b>	El usuario descarga el contenido desarrollado en el editor de texto, y almacenarlo en el servidor.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El usuario deberá de estar en una “Sesión”, para poder hacer uso de las funcionalidades de los editores de texto, tanto el grupal como el individual.
2	El sistema presentará la vista de la “Sesión” donde estará el editor de texto individual como el grupal, en los cuales habrá en cada una opción para “Bajar”(Descargar) lo contenido en el editor de texto.
3	El usuario elige la opción de “Bajar” lo que se encuentra en el editor de texto grupal.
4	El sistema procede a descargar el contenido en un archivo de texto, que se almacenará en el dispositivo de dicho usuario.

5	Finalmente el usuario tendrá que realizar los pasos del caso de uso “Almacenar Archivo”.
---	--

*Tabla 23. Descripción del caso de uso “Guardar el contenido del editor de texto”.*

<b>Nombre</b>	Crear Usuario
<b>Descripción</b>	El administrador del sistema crea un usuario nuevo para el sistema.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El administrador deberá de esta en la vista “Administración de Usuarios”, la cual es la vista principal de este tipo de usuario.
2	El sistema le presentará la vista “Administración de Usuario” en la cual se encuentran las siguientes opciones: <ul style="list-style-type: none"> <li>• Crear Usuario</li> <li>• Buscar Usuario</li> </ul>
3	El administrador ingresa a la vista de “Crear Usuario”
4	El sistema presentará una vista con una serie de campos a rellenar, en los cuales están los siguientes: <ul style="list-style-type: none"> <li>• Nombre Completo</li> <li>• Contraseña</li> <li>• Email</li> </ul> Y un botón de “Crear”, para finalizar la acción.
5	El administrador rellena los datos, y seleccionará el botón “Crear”.
6	El sistema presentará una ventana que indicará que la acción fue realizada correctamente.
<b>Excepciones</b>	
<b>Paso</b>	<b>Acción</b>
6	En caso de ingresar mal los datos, se presenta una ventana la cual indicará que deberán de ingresarse nuevamente dichos datos.

*Tabla 24. Descripción del caso de uso “Crear Usuario”.*

<b>Nombre</b>	Modificar Usuario
<b>Descripción</b>	El administrador del sistema modifica a un usuario ya insertado en el sistema.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El administrador deberá de esta en la vista “Administración de Usuarios”, la cual es la vista principal de este tipo de usuario.
2	El sistema le presentará la vista “Administración de Usuario” en la cual se encuentran las siguientes opciones: <ul style="list-style-type: none"> <li>• Crear Usuario</li> <li>• Buscar Usuario</li> </ul>
3	El administrador ingresa a la vista de “Buscar Usuario”.
4	El sistema presentará una vista donde habrá un campo de texto, para buscar a un usuario por Email y un botón “buscar” para realizar la acción de búsqueda.
5	El administrador ingresa el email del usuario al que quiere buscar y selecciona el botón “buscar”.
6	En caso de ingresar un email de un usuario ya creado en el sistema, se presentará el perfil del usuario con los datos presentados en campos de texto editable, y con un apartado de dos botones los cuales son “editar” y “eliminar”, dichos campos son: <ul style="list-style-type: none"> <li>• Nombre Completo</li> <li>• Contraseña</li> <li>• Email</li> <li>• Permiso</li> <li>• Pad_ID</li> </ul>
7	El administrador editará los campos de texto que sean de su elección y seleccionará el botón “editar”.
8	El sistema presentará una ventana que indicará que la acción fue realizada correctamente.
<b>Excepciones</b>	
<b>Paso</b>	<b>Acción</b>
6	En caso de escribir un email incorrecto, el sistema informará al administrador que dicho email no existe en los registros del sistema.

*Tabla 25. Descripción del caso de uso “Modificar Usuario”.*

<b>Nombre</b>	Eliminar Usuario
<b>Descripción</b>	El administrador del sistema elimina a un usuario ya insertado en el sistema.
<b>Secuencia normal</b>	
<b>Paso</b>	<b>Acción</b>
1	El administrador deberá de estar en la vista “Administración de Usuarios”, la cual es la vista principal de este tipo de usuario.
2	El sistema le presentará la vista “Administración de Usuario” en la cual se encuentran las siguientes opciones: <ul style="list-style-type: none"> <li>• Crear Usuario</li> <li>• Buscar Usuario</li> </ul>
3	El administrador ingresa a la vista de “Buscar Usuario”.
4	El sistema presentará una vista donde habrá un campo de texto, para buscar a un usuario por Email y un botón “buscar” para realizar la acción de búsqueda.
5	El administrador ingresa el email del usuario al que quiere buscar y selecciona el botón “buscar”.
6	En caso de ingresar un email de un usuario ya creado en el sistema, se presentará el perfil del usuario con los datos presentados en campos de texto editable, y con un apartado de dos botones los cuales son “editar” y “eliminar”, dichos campos son: <ul style="list-style-type: none"> <li>• Nombre Completo</li> <li>• Contraseña</li> <li>• Email</li> <li>• Permiso</li> <li>• Pad_ID</li> </ul>
7	El administrador seleccionará el botón “eliminar”.
8	El sistema presentará una ventana que indicará que la acción fue realizada correctamente.
<b>Excepciones</b>	
<b>Paso</b>	<b>Acción</b>
6	En caso de escribir un email incorrecto, el sistema informará al administrador que dicho email no existe en los registros del sistema.

*Tabla 26. Descripción del caso de uso “Eliminar Usuario”.*

2.9.3.2.3 *Diagrama de Secuencia*

En los siguientes diagramas (Diagramas 16 - 25) se presenta Diagramas de Secuencia correspondiente al Sprint 3.

En el siguiente diagrama (Diagrama 16) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Almacenar Archivo”.

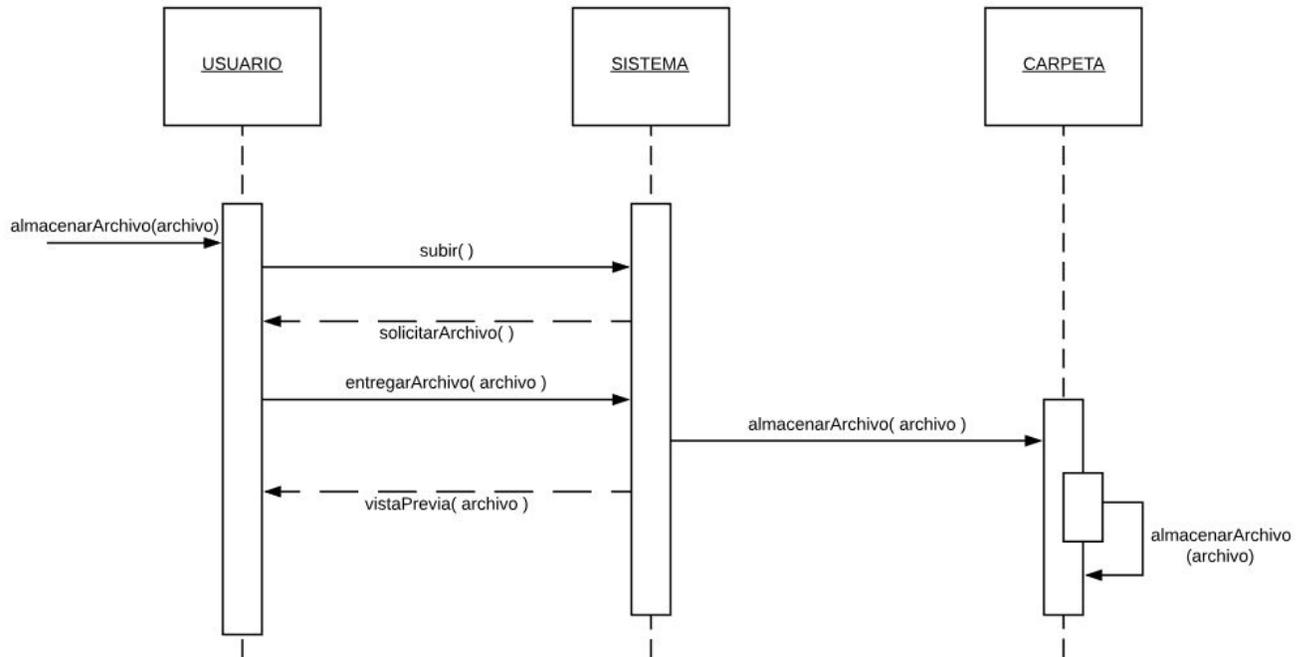


Diagrama 16. Diagrama de Secuencia “Almacenar Archivo”

En el siguiente diagrama (Diagrama 17) se presenta el Diagrama de Secuencia correspondiente al caso de uso “*Eliminar Archivo*”.

*Diagrama 17. Diagrama de Secuencia “Eliminar Archivo”*

En el siguiente diagrama (Diagrama 18) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Otorgar Permisos Carpeta”.

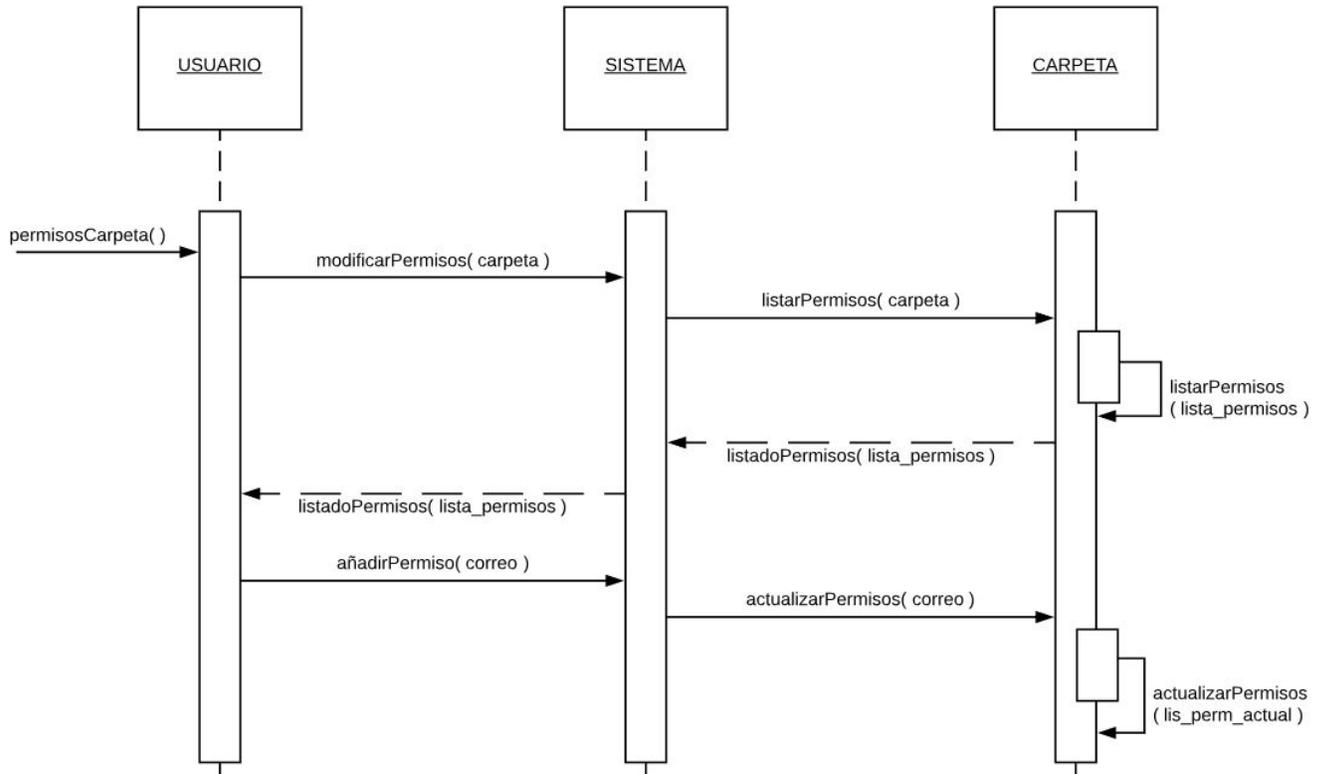


Diagrama 18. Diagrama de Secuencia “Otorgar Permisos Carpeta”

En el siguiente diagrama (Diagrama 19) se presenta el Diagrama de Secuencia correspondiente al caso de uso “Quitar Permisos Carpeta”.

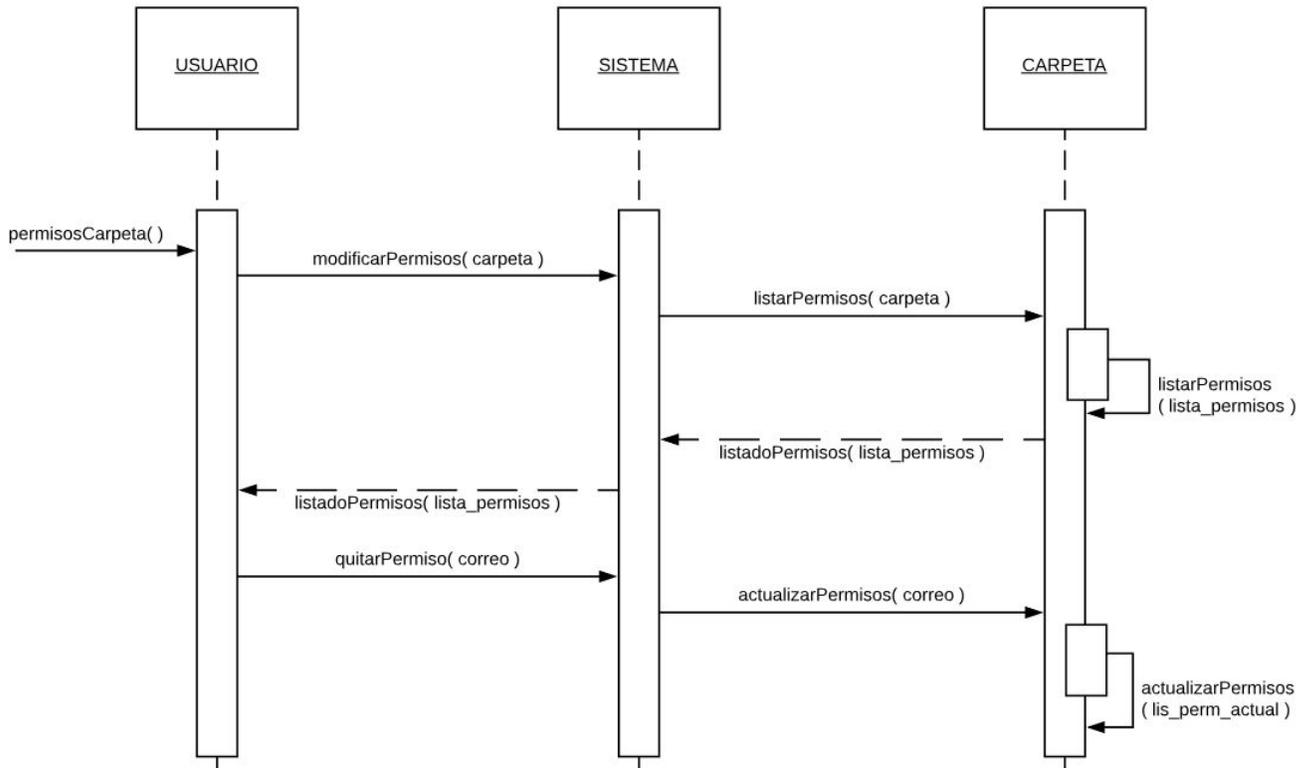


Diagrama 19. Diagrama de Secuencia “Quitar Permisos Carpeta”

En el siguiente diagrama (Diagrama 20) se presenta el Diagrama de Secuencia correspondiente al caso de uso “*Guardar Contenido Editor de Texto*”.

*Diagrama 20. Diagrama de Secuencia “Guardar Contenido Editor de Texto”*

En el siguiente diagrama (Diagrama 21) se presenta el Diagrama de Secuencia correspondiente al caso de uso “*Crear Usuario*”.

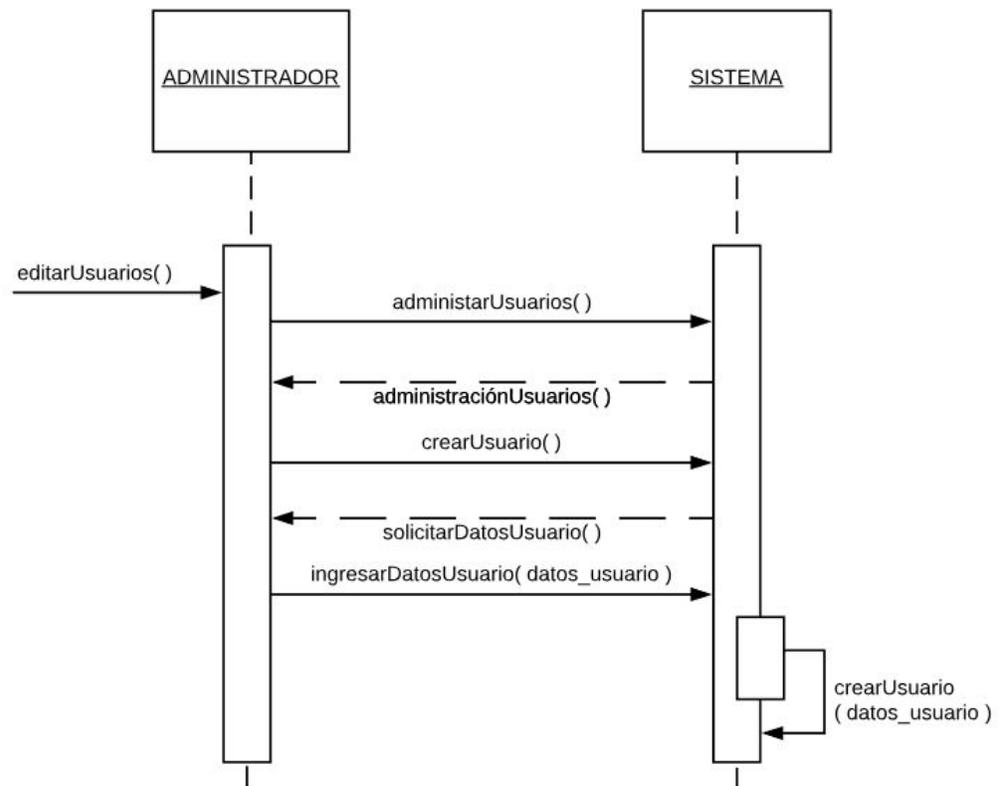


Diagrama 21. Diagrama de Secuencia “*Crear Usuario*”

*datos\_usuarios:*

- *Nombre Completo*
- *Correo*
- *Contraseña*

En el siguiente diagrama (Diagrama 22) se presenta el Diagrama de Secuencia correspondiente al caso de uso “*Modificar Usuario*”.

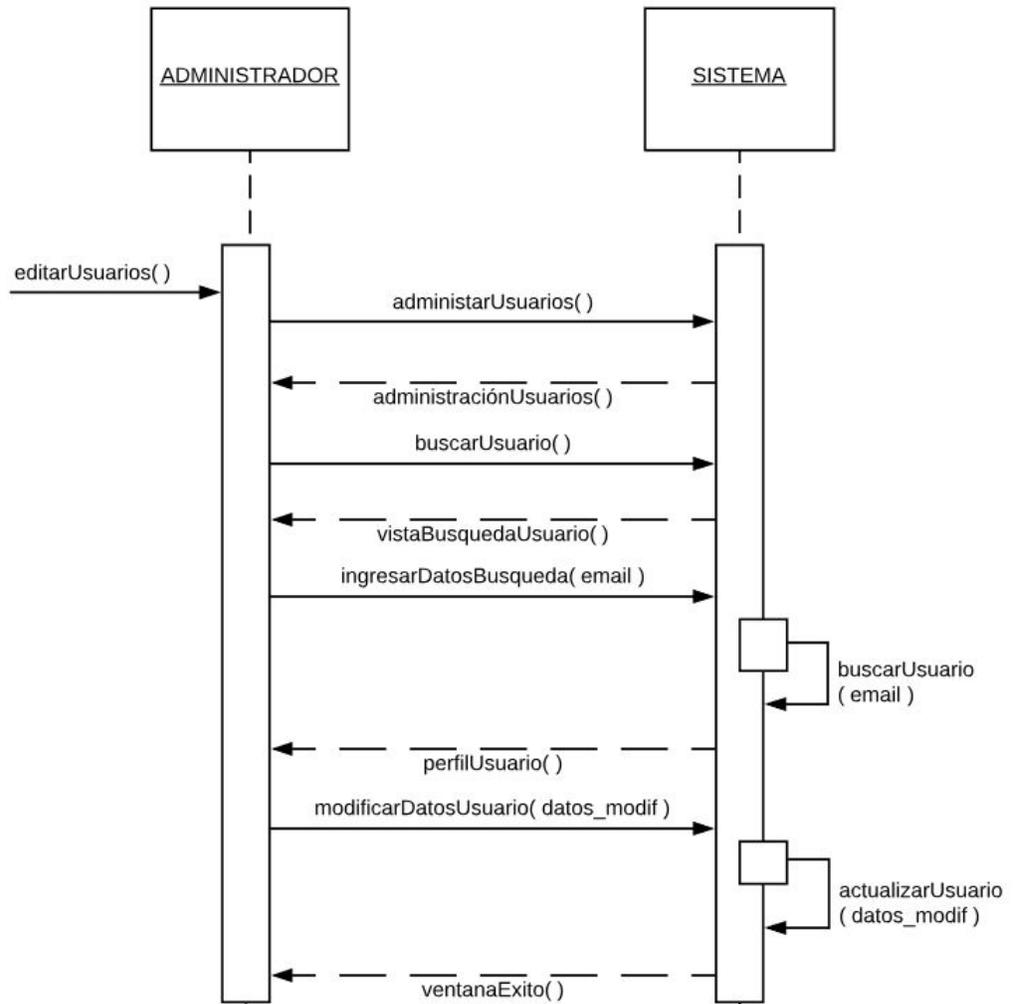


Diagrama 22. Diagrama de Secuencia “*Modificar Usuario*”

En el siguiente diagrama (Diagrama 23) se presenta el Diagrama de Secuencia correspondiente al caso de uso "Eliminar Usuario".

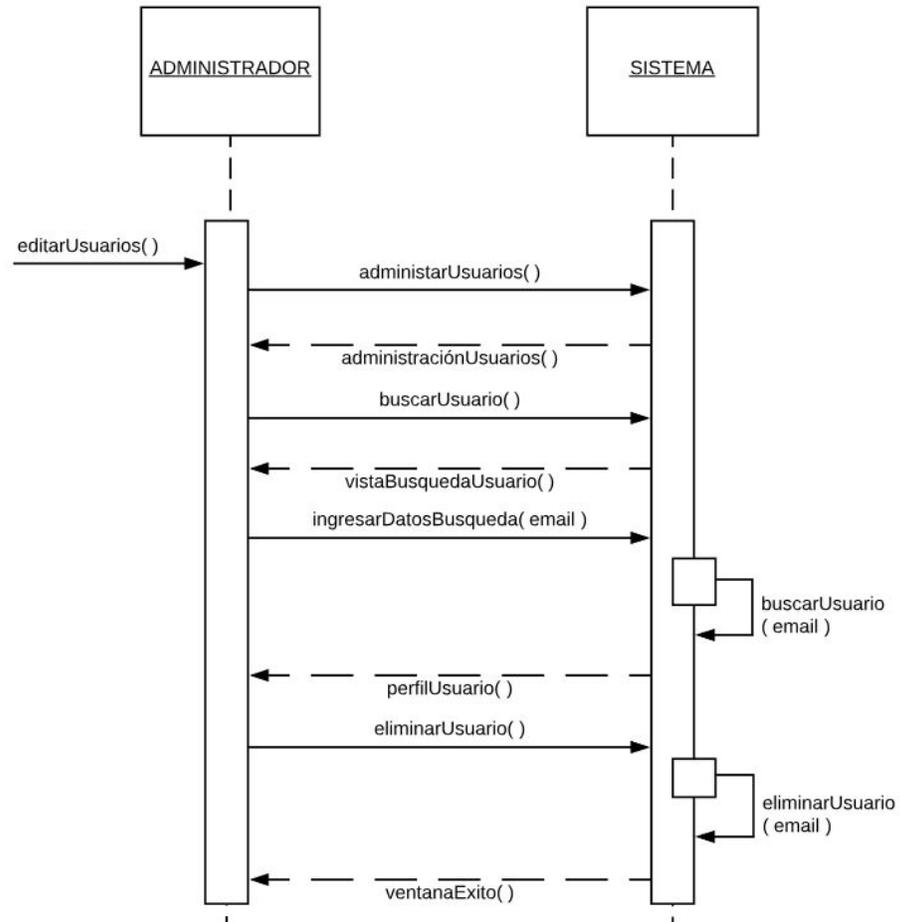


Diagrama 23. Diagrama de Secuencia "Eliminar Usuario"

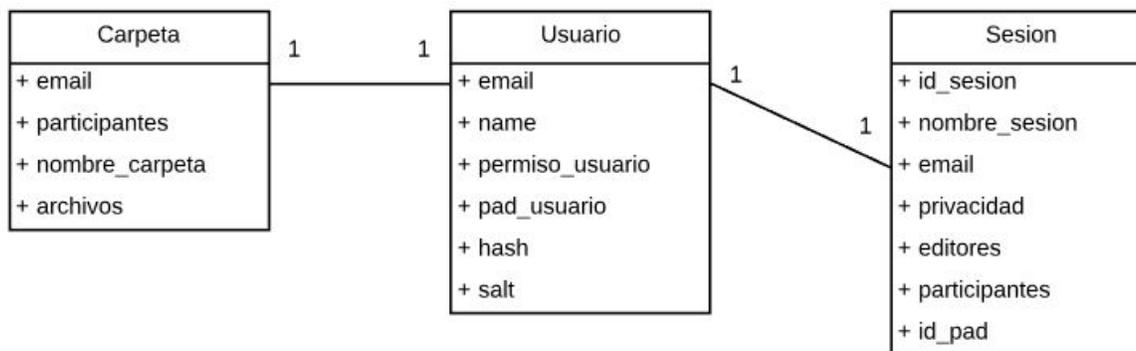
#### 2.9.3.2.4 Diagrama de Clases

A continuación se presenta el diagrama de clases (diagrama 24) concebido desde el Sprint 1, modificándose para el Sprint 2 y finalmente siendo retocado en el Sprint 3, donde se destaca que se añadieran los atributos “Permiso” y “Pad\_ID”, donde “Permiso” servirá para definir las acciones que podrá realizar dicho usuario, y “Pad\_ID” es la identificación del editor de texto perteneciente únicamente para dicho usuario.

*Diagrama 24. Diagrama de Clases “Diagrama de Clases Sprint 3”*

#### 2.9.3.2.5 Modelado de Base de Datos

A continuación se presenta el diagrama (Diagrama 25) del modelado de la base de datos para el proyecto, concebido desde el Sprint 1 y el cual cambio en el Sprint 3, por la relación entre el Usuario y la Carpeta, la cual es 1 a 1.

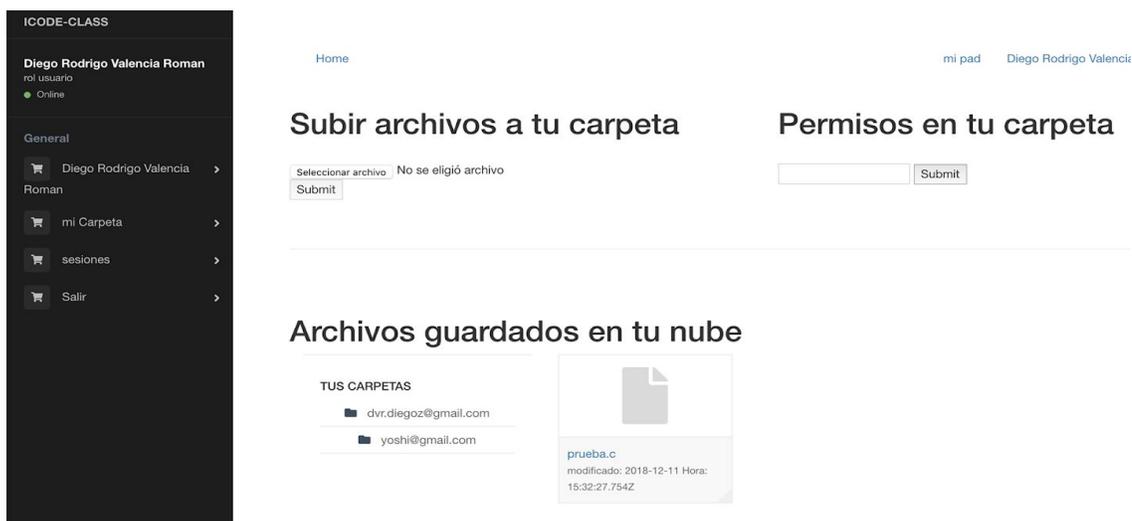


*Diagrama 25. Diagrama de Base de Datos “Diagrama de Base de datos Sprint 3”*

### 2.9.3.3 Implementación

A continuación se presenta en las siguientes ilustraciones (Ilustración 7, 8, 9, 10, 11) un avance preliminar de las vistas relacionadas al Sprint 3, es decir, relacionada al manejo de archivos y usuarios.

*Ilustración 7. Vista preliminar de la pantalla de Perfil de Usuario*



*Ilustración 8. Vista preliminar de la pantalla de Archivos en Carpeta*

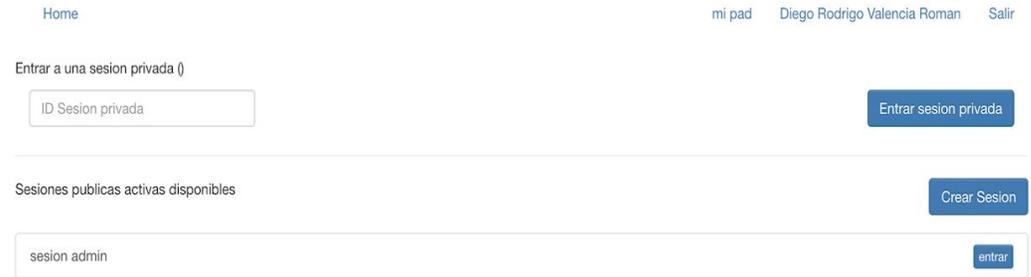
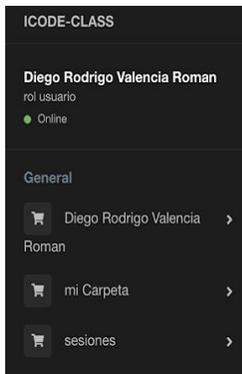


Ilustración 9. Vista preliminar de la pantalla de Sesiones

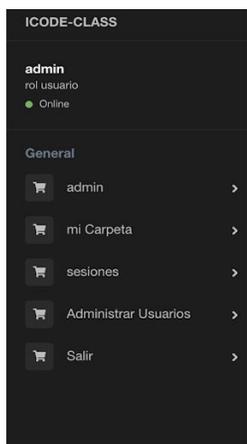


Ilustración 10. Vista preliminar de la pantalla de Registro de Usuario

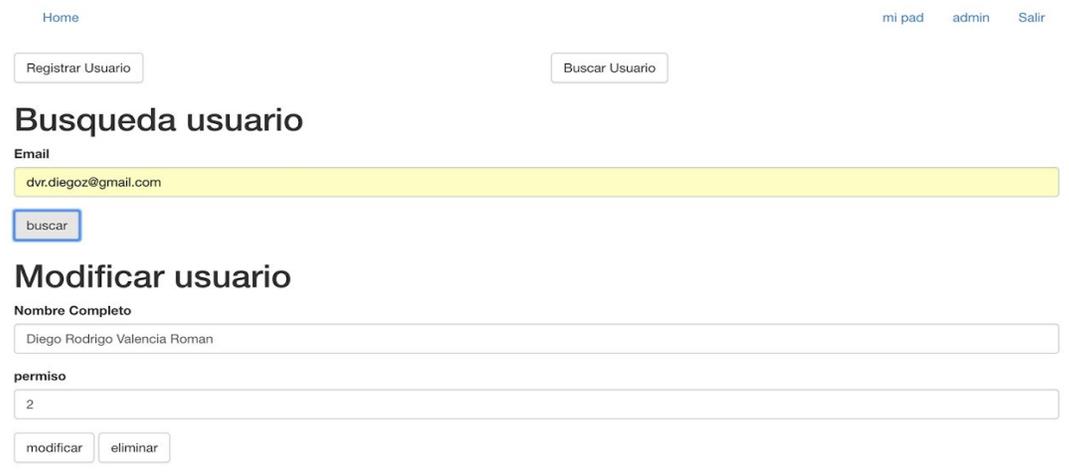
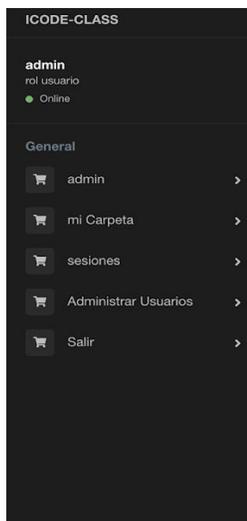


Ilustración 11. Vista preliminar de la pantalla de Modificación de Usuario

### 2.9.3.4 Aspectos Generales

#### 2.9.3.4.1 Responsables por actividad

<b>Sprint 3</b>	
<b>Actividad</b>	<b>Responsable</b>
Desarrollar la versión final del editor colaborativo	Daniela Oñate Manuel Tapia Diego Valencia
Implementar el marcado de sintaxis por lenguaje	Daniela Oñate Manuel Tapia Diego Valencia
Diseñar e implementar las vistas del editor en el sistema web	Daniela Oñate Manuel Tapia Diego Valencia
Pruebas del sistema web con el editor incorporado, utilizando sesiones	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Integración de todos los módulos del sistema	Daniela Oñate Manuel Tapia Diego Valencia
Implementar una herramienta dirigida al manejo de archivos	Daniela Oñate Manuel Tapia Diego Valencia
Definir la estructura y funcionamiento del manejo de archivos y carpetas	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Definir el tipo de archivos y carpetas del sistema	Daniela Oñate Manuel Tapia Diego Valencia
Definir los privilegios y restricciones de los archivos y carpetas del sistema	Daniela Oñate Manuel Tapia Diego Valencia
Diseñar el sistema de manejo de archivos y carpetas	Daniela Oñate Manuel Tapia Diego Valencia

Implementar la base de datos enfocada al manejo de archivos	Daniela Oñate Manuel Tapia Diego Valencia
Desarrollar e implementar el módulo de manejo de archivos en el servidor del sistema web	Daniela Oñate Manuel Tapia Diego Valencia
Diseñar e implementar las vistas de manejo de archivos del sistema web	Daniela Oñate Manuel Tapia Diego Valencia
Realizar una serie de Pruebas	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia
Presentación del Sprint final y Entrega del Sistema al Cliente	Luis Lecaros Daniela Oñate Charlotte Soto Manuel Tapia Fabián Urrutia Diego Valencia

*Tabla 27. Responsabilidades actualizadas para el Sprint 3*

## 2.9.3.4.2 Roles actualizados

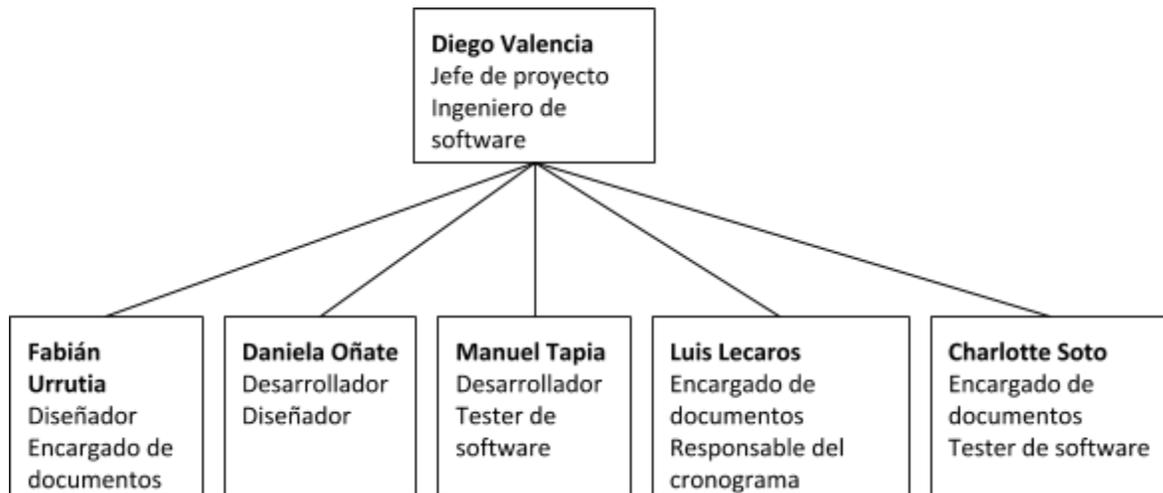


Figura 5. Roles Actualizados para el Tercer Sprint

### 2.9.3.4.3 Avance según Carta Gantt

En la siguiente figura (figura 6) se presentará el avance realizado según Carta Gantt para todo el desarrollo del proyecto el cual ha sido de un 93% hasta la fecha de hoy, completando todas las actividades que fueron planificadas para realizarse, quedando por completar el feedback con el cliente durante la presentación del Sprint Final, realizando las modificaciones pertinentes o bien entregando el producto final.

Cabe destacar que dentro de la realización de este Sprint se tuvo que realizar una reformulación de los tiempos debido a un atraso que ocurrió en el desarrollo de uno de los módulos por lo que se otorgó una semana más para finalizar con el desarrollo del proyecto y así, cumplir con la especificación de los requerimientos que fue firmada a un principio del desarrollo del proyecto.

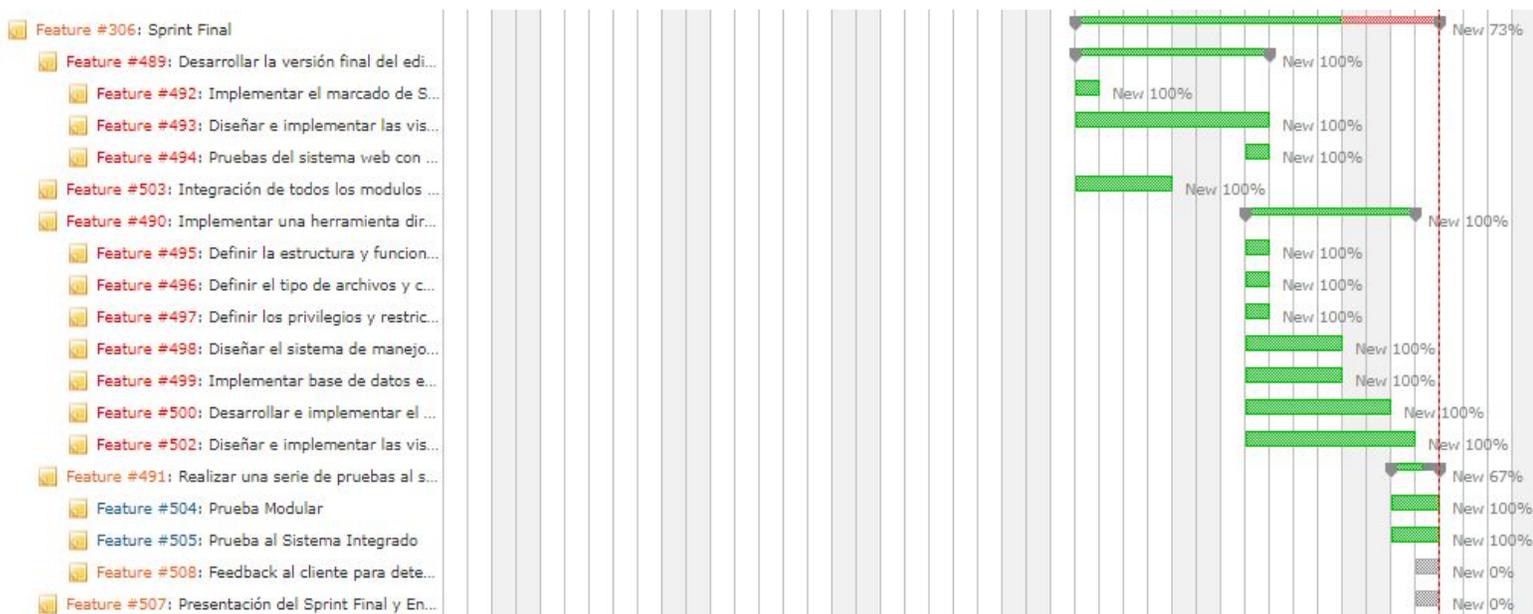


Figura 6. Avance de Carta Gantt hasta final del Último Sprint

### III. CONCLUSIONES

Una vez terminado el último Sprint, que da como finalizado el proyecto iCodeClass, se obtuvieron una serie de conclusiones, resultados de los sprints pasados y esta entrega final, tales fueron:

- No subestimar o bien sobrestimar las distintas actividades a realizar para un proyecto de software, ocurrió que en tareas o roles se hizo una distribución incorrecta, los cuales se demostraron en retrasos de las fechas, o en otros resultados no esperados.
- A pesar de los acuerdos llegados con el Cliente, no se pudo realizar una entrega total, o explícita de lo que el Cliente esperaba, aún así se cumplen de una manera indirecta, ya que se presentan los elementos principales con sus funciones principales, tales se pueden reflejar en los Requisitos referentes a las Carpetas, a base de ejemplo.

## IV. REFERENCIAS BIBLIOGRÁFICAS

- [1] Metodología SCRUM, [En línea]. Disponible en: <https://proyectosagiles.org/que-es-scrum>
- [2] Precios VTR, [En línea]. Disponible en: <https://vtr.com/productos/NegociosPacks/doble-pack-banda-ancha-television>
- [3] Precios Alimentación, [En línea]. Disponible en: <http://lanacion.cl/2018/08/02/alimentacion-laboral-cuanto-cuesta-almorzar-en-chile>

## V. ANEXOS

### Implementación (Código)

A continuación se muestra en la siguiente ilustración (Ilustración [12 - 14]) los métodos de código perteneciente al módulo de autenticación del usuario, tanto de lado del cliente como del lado del servidor y sus respectivas vistas del sistema web o producto.

```
public isLoggedIn(): boolean {
  const user = this.getUserDetails();
  if (user) {
    return user.exp > Date.now() / 1000;
  } else {
    return false;
  }
}

private request(method: 'post'|'get', type: 'login'|'register'|'profile', user?: TokenPayload): Observable<any> {
  let base;

  if (method === 'post') {
    base = this.http.post(`http://localhost:3000/api/${type}`, user);
  } else {
    base = this.http.get(`http://localhost:3000/api/${type}`, { headers: { Authorization: `Bearer ${this.getToken()}` } });
  }

  const request = base.pipe(
    map((data: TokenResponse) => {
      if (data.token) {
        this.saveToken(data.token);
      }
      return data;
    })
  );

  return request;
}

public register(user: TokenPayload): Observable<any> {
  return this.request('post', 'register', user);
}

public login(user: TokenPayload): Observable<any> {
  return this.request('post', 'login', user);
}
```

*Ilustración 12. Código lado cliente*

```
module.exports.login = function(req, res) {  
  
  // if(!req.body.email || !req.body.password) {  
  //   sendJSONresponse(res, 400, {  
  //     "message": "All fields required"  
  //   });  
  //   return;  
  // }  
  
  passport.authenticate('local', function(err, user, info){  
    var token;  
  
    // si passport catch un error  
    if (err) {  
      res.status(404).json(err);  
      return;  
    }  
  
    // si se encuentra usuario  
    if(user){  
      token = user.generateJwt();  
      res.status(200);  
      res.json({  
        "token" : token  
      });  
    } else {  
      // si no se encuentra usuario  
      res.status(401).json(info);  
    }  
  })(req, res);  
  
};
```

Ilustración 13. Código lado servidor

```
module.exports.register = async function(req, res) {  
  
  // if(!req.body.name || !req.body.email || !req.body.password) {  
  //   sendJSONresponse(res, 400, {  
  //     "message": "All fields required"  
  //   });  
  //   return;  
  // }  
  
  var user = new User();  
  var id;  
  var args = {  
    padID: req.body.email  
  }  
  await etherpad.createPad(args,(error,data)=>{  
    if(error){  
      console.error('Error al crear pad: ' + error.message);  
      if (error.message = "padID does already exist") {  
        id=req.body.email;  
      }  
    }  
    else{  
      console.log('New pad created: ' + data);  
      console.log(id);  
    }  
  });  
  console.log(id +'id antes de user');  
  user.pad_usuario = req.body.email;  
  console.log(user.pad_usuario+' pad usuario');  
  user.name = req.body.name;  
  user.email = req.body.email;  
  //perimos 0 solo permite entrar a sesiones  
  //permiso 1 permite al usuario crear sesiones, mas lo del permiso 0  
  //permiso 2 permite al usuario tener una carpeta y guardar archivos, mas lo de permiso 0  
  //permiso 3 permite al usuario hacer todo lo anterior  
  user.permiso_usuario = 0;  
  
  user.setPassword(req.body.password);  
  
  user.save(function(err){  
    var token;  
    token = user.generateJwt();  
    res.status(200);  
    res.json({  
      "token" : token  
    });  
  });  
}
```

Ilustración 14. Código lado del servidor

A continuación se muestra en la siguiente ilustración (Ilustración [15 - 20]) los métodos de código perteneciente al módulo de administración de usuario, administración de archivo y administración de sesiones.

```
module.exports.modificarUsuario = function(req,res) {
  var cambios = {
    name:req.body.nombre,
    permiso_usuario:req.body.permiso_usuario
  }

  User.update({email:req.body.email},{ $set:cambios},(err,doc)=>{
    if (err) {
      console.log(err);
    }else {
      res.status(200);
      res.send(doc);
    }
  });
}

module.exports.eliminarUsuario = function(req,res){
  User.deleteOne({ email: req.body.email }, function (err,doc) {
    if (err){
      return handleError(err)
    }else{
      Carpeta.deleteOne({email:req.body.email},(err,response)=>{
        if (err) {
          console.log(err);
        }else{
          res.status(200);
          res.send(response);
        }
      });
    }
  });
}
```

*Ilustración 15. Método Modificar Usuario y Eliminar Usuario*

```
module.exports.obtenerUsuario = function(req, res) {
  User.find({email:req.body.email},(err,doc)=>{
    if (err) {
      console.log(err);
    }else{
      res.status(200);
      res.send(doc);
    }
  });
}

//CarpetaArchivos

module.exports.eliminarArchivo=function(req,res) {
  Carpeta.update( {email:req.body.email} , {"$pull": { "archivos" : {"_id":req.body._id} }},(err,doc)=>{
    if (err) {
      console.log(err);
    }else{
      res.status(200);
      res.send(doc);
    }
  });
}

module.exports.guardarArchivo = function(req, res) {
  var infoarchivo = {Nombre:req.body.nombre_archivo,data:req.body.file,modificacion:req.body.modificacion}
  console.log(infoarchivo);
  console.log(req.body.email);
  Carpeta.update({email: req.body.email},
    { $push: { archivos: infoarchivo } } ,
    (err,response)=>{
    if (err) {
      console.log(err);
    }else{

```

*Ilustración 16. Método Obtener Usuario, Eliminar Archivo y Guardar Archivo*

```
module.exports.obtenerCarpeta = function(req,res) {
  var respu = [];
  var aux =[];
  /* Carpeta.find({participantes:req.body.email },(err,doc)=>{
    console.log('consulta0');
    console.log(doc);

  });*/
  Carpeta.find({ email: req.body.email },(err,response)=>{
    if (err) {
      console.log(err);
    }else{
      console.log('primera consulta');
      console.log(response)
      aux = aux.concat(response);
      Carpeta.find({participantes:req.body.email },(err1,doc)=>{
        if (err1) {
          console.log(err1);
        }else{
          console.log('segunda consulta');
          console.log(doc);
          aux = aux.concat(doc);
          res.status(200);
          res.send(aux);
        }
      });
      console.log('sender final');
      console.log(aux);
      /*res.status(200);
      res.send(aux);*/
    }
  });
}
```

*Ilustración 17. Método Obtener Carpeta*

*Ilustración 18. Método Dar Permiso Carpeta y Entrar Sesión*

*Ilustración 19. Método Modificar Permiso Sesión*

*Ilustración 20. Método Obtener Sesión, Eliminar de Sesión y Obtener Sesiones*